Advances in Computational Design, *Vol. 4, No. 2 (2019) 155-177*
DOI: https://doi.org/10.12989/acd.2019.4.2.155

Symbol recognition using vectorial signature matching for building mechanical drawings

Chi Yon Cho*, Xuesong Liu[a] and Burcu Akinci[b]

Department of Civil and Environmental Engineering, Carnegie Mellon University,
5000 Forbes Avenue, Pittsburgh, PA 15213, USA

(Received September 29, 2018, Revised December 25, 2018, Accepted February 14, 2019)

Abstract. Operation and Maintenance (O&M) phase is the main contributor to the total lifecycle cost of a building. Previous studies have described that Building Information Models (BIM), if available with detailed asset information and their properties, can enable rapid troubleshooting and execution of O&M tasks by providing the required information of the facility. Despite the potential benefits, there is still rarely BIM with Mechanical, Electrical and Plumbing (MEP) assets and properties that are available for O&M. BIM is usually not in possession for existing buildings and generating BIM manually is a time-consuming process. Hence, there is a need for an automated approach that can reconstruct the MEP systems in BIM. Previous studies investigated automatic reconstruction of BIM using architectural drawings, structural drawings, or the combination with photos. But most of the previous studies are limited to reconstruct the architectural and structural components. Note that mechanical components in the building typically require more frequent maintenance than architectural or structural components. However, the building mechanical drawings are relatively more complex due to various type of symbols that are used to represent the mechanical systems. In order to address this challenge, this paper proposed a symbol recognition framework that can automatically recognize the different type of symbols in the building mechanical drawings. This study applied vector-based computer vision techniques to recognize the symbols and their properties (e.g., location, type, etc.) in two vector-based input documents: 2D drawings and the symbol description document. The framework not only enables recognizing and locating the mechanical component of interest for BIM reconstruction purpose but opens the possibility of merging the updated information into the current BIM in the future reducing the time of repeated manual creation of BIM after every renovation project.

Keywords: Building Information Modeling (BIM); Facility Management (FM); symbol recognition; vectorial signature matching; building mechanical drawings

1. Introduction

The potential benefit of using BIM for Facility Management (FM) is improved tradespeople performance, supported by efficient document management, equipment localization and integration of building asset data (Cho *et al*. 2018a). However, due to the low adoption of BIM in

*Corresponding author, Ph.D. Candidate, E-mail: chiyonc@andrew.cmu.edu
[a] Ph.D., E-mail: pine@cmu.edu
[b] Professor, E-mail: bakinci@cmu.edu

http://www.techno-press.org/?journal=acd&subpage=7 ISSN: 2383-8477 (Print), 2466-0523 (Online)

the design of MEP systems and lack of standards on producing FM-ready BIM, commercial building owner and facility managers are still in an environment that cannot utilize MEP BIM in the O&M phase. The key barrier to adoption is the creation of BIM for FM (McArthur 2015). A previous study has shown that one can save up to 95% of the modeling time by automating the BIM creation process (Bortoluzzi *et al*. 2017).

The authors previously proposed a reconstruction approach that could generate BIM automatically from 2D building mechanical drawings to lower the barrier of adopting BIM for FM (Cho and Liu 2017). In previous work, various mechanical components that need to be recognized for building model reconstruction were categorized and a prototype was developed that reconstructs the duct components using the minimum circuit finding approach.

A major challenge that the authors have identified in the previous study is that many mechanical symbols needed to be recognized from the building mechanical drawings. Compared to architecture and structure drawings, which primarily use lines and texts to depict the building elements such as walls, windows, columns and beams, mechanical drawings contain a variety of symbols to represent different types of equipment and assets, including variable air volume (VAV) box, air handling unit (AHU), diffusers, registers, dampers, sensors (e.g., temperature, humidity and smoke detector), fans and so on. In order to recognize these symbols from the building mechanical drawings, the authors first investigated the different approaches regarding symbol recognition and previous literatures that have explored such approaches. Section 2 summarized these research studies and discussed the multiple issues regarding 2D building mechanical drawings and current problems that need to be resolved. Based on previous studies, the authors proposed a framework utilizing a vectorial signature matching method to recognize symbols and the results are shown in Section 3. Lastly, it is followed by a conclusion in Section 4 including discussions and the future direction of this research.

2. Background

Over the past decades, symbol recognition techniques have been applied to various domains such as automatic recognition of logic circuit diagrams (Lin *et al*. 1985), architectural drawings (Dosch *et al*. 2000, Liu *et al*. 2017, Santos *et al*. 2011, Valveny and Martí 1999, Zhi *et al*. 2003), structural drawings (Lu *et al*. 2005, Lu and Lee 2017), line drawings (Ha and Eck 2017, Ventura and Schettini 1994) and composite graphics (Schettini 1996). These applications demonstrated that utilizing symbol recognition enables automated processing of image files or electronic drawings that are otherwise challenging to be handled manually. The following paragraphs reviewed the previous studies and symbol recognition techniques and compared them considering the unique characteristics of building mechanical drawings. Generally, vectorization is a common process if the input is a raster image for eliminating noise and unwanted features in symbol recognition. However, nowadays all buildings are constructed based on digital CAD format drawings and the architect is required to deliver the drawing documents to the owner (American Institute of Architects 1997, Fallon and Palmer 2007). Therefore, the authors focus on processing the vector format drawings such as Drawing Exchange Format (DXF) file.

In the Architecture, Engineering and Construction (AEC) domain, many studies applied symbol recognition techniques to architectural drawings and a couple of previous studies explored applying such techniques to structural drawings. However, to the best of the author’s knowledge, there is none for building mechanical drawings. An efficient 3D creation method suggested by

Santos *et al*. (2011) utilized building floorplans and photographs to semi-automatically generate a 3D model. The doors and windows were automatically detected using edge detection techniques and the texture information was extracted from photographs by detecting contours. Zhi *et al*. (2003) used a graph-based approach to identify the closed loops in architectural drawings by extracting geometrical and topological information. It utilized the attributes of room, corridor, wall, balcony, window and door to identify loops and unit functions to build an evacuation model. In architectural drawings, it is common that the floorplan contains closed-loops surrounded by walls. And the window and door elements are usually placed overlapping within the wall elements. In mechanical drawings, on the other hand, the ducts could be expressed either with one line or two lines. Moreover, the ducts could overlap with one another and many annotations including symbols are placed nearby. Hence, it is more difficult to represent the spatial and topologic relationship among mechanical components.

Lu and Lee (2017) developed a semi-automated approach that could recognize the grid related symbols and text information in structural drawings. The grid was defined based on the horizontally positioned alphabetic characters and the numbers on the vertical direction. Using the grid information and the text information obtained through the Optical Character Recognition (OCR) algorithm together, the structural components were located. In Lu *et al*. (2005), a hierarchical recognition model was proposed that could extract dimension, grid, column and steel structure information from Construction Structural Drawings (CSD). Walls, column and internal steel components were reconstructed using section tracking, segment recognition and component relation graph methods. In both studies (Lu *et al*. 2005, Lu and Lee 2017), grid information was extracted from the structural drawings to identify the location of the structural components. This is a unique characteristic for structural drawings because the load-bearing structural elements such as beams, columns and trusses are usually located along the grid lines. This method is not applicable for recognizing symbols in mechanical drawings because the symbols are not located according to certain horizontal lines or vertical lines.

Extracting information from digital CAD drawings were also studied in other domains such as robots (Yin *et al*. 2013) and machines (Ahmad and Haque 2001). Although the approaches are domain dependent, the studies have shown that geometric features in drawings could be translated into functional meanings of certain structures or direction based on relationship logic and domain-specific knowledge.

2.1 Symbol recognition techniques

The symbol recognition process is generally divided into three phases: Representation phase, Description phase and the Classification phase. The representation phase includes the preprocessing steps that aim to reduce noise and the amount of data which outlines and represents the input component (Cordella and Vento 1999). For example, an input can be represented as pixels, curves, features, primitives, or samples (Jain *et al*. 2000). In the description phase, the boundary and region of the symbol of interest in the input are described. A description can be said to be an aspect of the representation. Let's assume one is trying to distinguish between strawberries and bananas in a raster image. In this case, the representation of both fruits will be in a form of a pixel and the description will be the color and shape. Generally, for technical drawings, topological or geometric features are used for description since the shape of symbols are made based on the combination of simple geometric shapes (Cordella and Vento 1999). Finally, the classification phase is the step where the symbols are recognized and classified. In other words,

Table 1 Classification of symbol recognition techniques by phases (from Cho *et al*. 2018b)

Phases	Techniques	Relevant techniques
Representation phase	Connected-component labelling, Distance transform, Hough transform, Internal and/or external contours, Run-length based representations, Thinning and polygonal approximation, Thin-thick separation, Wavelet transform, Fourier transform, Pixel level constraints, Auto-Regressive (AR), Shape context, Shape matrix, Angular Radial Transform (ART), BSM	Connected Component Labelling, Distance transform, Internal and/or external contours, Shape context, Shape matrix
Description phase	Fourier descriptors, Geometric and topologic features, Structural descriptors, Syntactic descriptors, Moment invariants (Zernike, Legendre), Hybrid descriptors	Geometric and topologic features, Structural descriptors, Hybrid descriptors
Classification phase	Bayes plug-in, Decision tree, Fisher linear discriminant, Graph matching, Heuristic techniques, Logistic classifier, Multi-layer perceptron, Nearest mean classifier, Neural network classifier, Parzen classifier, Radial basis network, Statistical classifier, Subspace method, Support vector classifier, Syntactic parser, Template matching	Decision tree, Graph matching, Template matching, Neural network classifier, Statistical classifier, Support Vector Machine classifier, Signature matching

classification indicates the set of methods that allow the symbol recognition (Tabbone and Terrades 2014).

In addition, the approaches of symbol recognition could be classified based on the matching feature property or comparing decomposed parts: Statistical-based, Structural-based, Hybrid and Syntactic. Statistical based recognition methods describe the statistical property of the extracted feature and the structural based recognition methods focuses on meaningful regions or geometric primitives and their relations and stores the property information into structures (e.g., graphs, tree, or network of constraint) (Zhang and Liu 2007). The hybrid method extracts statistical properties based on the structural representations of symbols (Dosch and Lladós 2003). Lastly, the syntactic methods are related to syntax analysis which extracts the features of a sentence in languages.

Furthermore, based on how the shape of symbols are represented and described, the approaches could be categorized into two classes: Contour-based and Region-based. The contour-based method focuses on the shape boundary information and not the internal shape content. Accordingly, it is sensitive to noise and variations because only a portion of the shape information is used (Zhang and Lu 2004). On the other hand, the region-based method utilizes all available shape information. Thus, it is generally more robust and provides accurate retrieval of the shape information.

The authors have reviewed the previously used symbol recognition techniques by phases based on multiple review papers (Cordella and Vento 1999, Jain *et al*. 2000, Tabbone and Terrades 2014) and classified each technique by phases in Table 1. Furthermore, the relevant techniques (or candidate methods) for this study were filtered out on the rightmost column in Table 1.

Note that the majority of the previously used symbol recognition techniques handles pixel-based raster images. Thus, it was vague when searching for which symbol recognition techniques to select to recognize symbols in the 2D building mechanical drawings which is a vector-based input. One of the reason is that there are relatively fewer studies regarding vector-based images due to lack of publicly available vector drawings datasets (Ha and Eck 2017). Therefore, the authors explored each technique mentioned in Table 1 and excluded non-relevant techniques from consideration. The following paragraphs give a brief description of the techniques and explain why they are relevant or non-relevant for this study.

Connected Component Labelling (CCL) is used to detect connected components or regions. This method is applied in graph theory where the subsets of connected components are uniquely labeled. CCL was commonly used in previous studies (Ahmed *et al*. 2011, De *et al*. 2014, Maity *et al*. 2017) to recognize objects in the AEC domain drawings. Accordingly, this technique is relevant because symbols in 2D building mechanical drawings can also be represented by connected parts such as the duct component and diffusers.

Fourier transform and wavelet transform techniques are widely used in signal processing. It extracts the important part of the signal which is the coefficients. Lines or arcs can be detected using transform-based techniques but they are non-relevant since the vector-based drawings already contain geometric entities such as lines, circles and arcs. For the same reason, Hough transform and thinning techniques are also classified as non-relevant. The Auto-Regressive (AR) method is also used in signal processing which is a random process representation that describes the features related to time series data. Thus, AR is a non-relevant technique.

The run-length based method represents data that have the same values occurring consecutively. The authors classified this approach as a non-relevant technique because there are different symbols in the building mechanical drawings that have the same sequence of elements.

Moment invariants could be used to extract a set of invariant features from images with a different view of the same type of object. The approach is known to perform well on contour-based shapes but poorly for region-based shapes which have interior content (Zhang and Lu 2004). Thus, it is suitable for describing simple shapes. This method is non-relevant since the symbols in 2D building mechanical drawings do have interior content and do not have multiple viewpoints.

Based on comprehensive review of previous symbol recognition techniques, the relevant techniques are the following: Connected Component Labelling (CCL), Distance transform, Internal and/or external contours, Shape context and Shape matrix for the representation phase and Geometric and topologic features, Structural descriptors and Hybrid descriptors for the description phase.

Although CCL is categorized as a relevant technique considering that it is an algorithmic application of graph theory, it is a commonly used technique on pixel-based images to detect regions. The algorithm traverses the pixels and labels them based on the connectivity and relative pixel values of their neighbors. If the concept of CCL is to be applied in vector-based input, multiple points would need to be generated along the path of the vector lines. For recognition, graph matching technique would be applied but the computational complexity is anticipated to be high.

The distance transform is an operation of converting a binary image to an image where each element is set to the distance of the nearest boundary contour (Borgefors 1984, Tabbone *et al*. 2006). If all the vectors in the vector-based input are sampled enough and converted to feature points, the distances can be defined from the non-feature points. However, the proper extent of sampling the points along the vectors remains a question.

Shape context is a correspondence-based shape matching method that captures the distribution of the points at a reference point (Belongie *et al*. 2001). For each sample point along the internal and external contour of a shape, a log-polar histogram is obtained based on the length and orientation of the vectors. This histogram is the called the context and the histogram of each point is flattened and concatenated to form the shape context map (Zhang and Lu 2004). The shape context approach can be applied for vector-based inputs since finding the list of points on shape edges can be done by sampling the shape with uniform spacing between vectors. However, there are originally three components that are required to define the shape distance between shapes when measuring the shape similarity. The shape distance between shapes is defined as a weighted sum of three terms: shape context distance, image appearance distance and bending energy (Belongie *et al*. 2001). The term image appearance is related to textural information or color around the corresponding points which is not captured in the shape context. Bending energy is the transformation cost which is a measurement of the amount of necessary transformation to align the shapes. Among these three terms, the image appearance distance cannot be obtained in the vector-based input. Thus, if the shape context is to be applied, the performance of the symbol recognition would depend on the remaining two terms with unknown weights.

Shape matrix is obtained by a polar quantization of a shape (Goshtasby 1985). It captures both outer and inner geometry of the shape by overlaying a polar raster of concentric circles and radial lines in the center of the mass (Loncaric 1998). Then, the binary value of shape is sampled at the intersections of the circles and radial lines. This method can be applied to vector-based input if the points are sampled sufficiently along the vectors because the shape matrix is a sparse sampling method. However, shape matching using a shape matrix is easily affected by noise and it is expected to be expensive (Zhang and Lu 2004).

Most methods mentioned here regarding the representation phase are related to processing pixel-based images and not vector-based inputs. Note that the symbols in building mechanical drawings can be represented as geometric primitive features. Thus, it would be preferable to describe the geometric primitive features directly in the description phase instead of converting the vector-based input similarly to a raster image and applying the methods that are used for representing symbols in raster images.

Descriptors based on geometric and topologic features describes the symbol by a set of features of the elementary geometry structures and the relation between them. For example, symbol signature can be generated by the relation between the intersecting lines (e.g., number of diverging segments, angles between lines) and by numerical measures (e.g., length, orientation, sharp angles) of these lines (Schettini 1996). Then, the symbol recognition system can use this signature filter to identify the presence of a set of symbols that have that feature. This approach is suitable to describe symbols in vector-based input because the elements in vector-based input can be broken down into elementary vector structures.

Graphs are used as a structural descriptor since they are suitable for representing structural information (Cordella and Vento 1999). Especially, Attributed Relational Graphs (ARGs) are good support for structural descriptions which consists of nodes, edges and their associated attributes. Previously, ARG which is invariant to translation, scaling and rotation have been used to represent the graphical symbols in engineering drawings and the recognition was done based on the error-tolerant subgraph isomorphisms matching algorithm (Messmer and Bunke 1995). Another use case of a graph is the Region Adjacency Graphs (RAG) which was utilized to recognize symbols in architectural hand-drawn plans (Lladós *et al*. 2001). The difference between ARG and RAG is that the line segment forms the nodes and edges in ARG while the regions are represented as nodes and the neighboring relations between regions as edges in RAG. However, the drawbacks of the use of graph matching are that it is computationally complex and is sensitive to noise and distortion. For example, missing or added nodes and edges due to human error in 2D drawings may result in forming a distorted graph.

Hybrid descriptor such as the vectorial signature method extracts statistical information from the structural representation of shapes (Dosch and Lladós 2003, Zhang and Liu 2007). An advantage of using vectorial signatures is that it reduces the computation time in the symbol recognition phase and it allows to define regions of interest solving the symbol segmentation problem (Dosch and Lladós 2003). In addition, it is invariant to affine transformations such as rotation, scaling and translation. The proposed framework of this study is built upon this hybrid approach since the geometric primitives of the symbols in the building mechanical drawing can be represented based on the multiple relationships between geometric primitives.

Lastly, the candidate techniques that could possibly be used in the classification phase is quite clear based on the categorized relevant techniques for the representation and description phase: Decision tree, Graph matching, Template matching, Statistical classifier and Signature matching. Learning based algorithms such as Support Vector Machine (SVM) or neural network methods could also be used if one has a significant amount of data that enables training the data sets.

2.2 Utilized symbol recognition techniques in previous studies in the AEC domain

In this section, the authors reviewed the types of symbol techniques that were utilized in previous studies particularly in the AEC domain (Table 2). In Maity *et al*. (2017), 'callout' symbols which contain destination sheet names represented in circular shapes were detected in the AEC documents to improve the accessibility of the information when navigating through a large number of document sheets. The Connected Component Labelling (CCL) technique was used to remove the unwanted parts during the process and eventually recognized the circle shapes using the positional features and size-based thresholding technique. For text recognition, OCR method with the SVM as a classifier was utilized. This method is specifically designed to recognize the callout symbols in the AEC document. It is limited to recognize circular shapes which are not applicable to recognize the various type of symbols that are in the building mechanical drawings.

Lines in engineering drawings can be detected by sampling points at certain intervals of pixels and testing collinearity using an incremental algorithm (De *et al*. 2014). In De *et al*. (2014), arcs and circles were detected using the component labeling approach and the arrowhead was detected using the distance transform values.

A deformable template matching model was proposed by Valveny and Martí (1999) which could recognize lineal symbols in hand-drawn architectural drawings. Unlike other symbol recognition studies, this method does not require the process of vectorization and feature extraction which are commonly applied to raster images. Since hand-written drawings contain noise and distortion, this approach aims to match the symbols by finding the least possible deformation of a symbol that best fits to the input image by using a Bayesian probabilistic framework. However, this approach is limited to recognize lineal symbols represented with straight lines.

In Ahmed *et al*. (2011), text/graphics segmentation approach was proposed as a useful pre-processing step for document analysis. To extract and separate the two layers of graphical information and textual information in the architectural floor plans, the walls were first detected and removed prior to segmentation by successive morphological binary erosion followed by successive morphological binary dilation. As a result, only the thick components remain including the thick walls and thick characters such as the title text of the floor plan.

Ah-Soon and Tombre (2001) proposed a method for recognizing architectural symbols based on the description of the network of constraints on geometrical features. Two types of constraints were applied to the segments of the feature: connection constraints which describe the connection relations between segments and simple constraints which is a description of the size of its feature. Then the search of symbols is done through the propagation of the segments in the network. This approach is different from matching methods since it searches all the features separately and verify

Table 2 Previously used symbol recognition techniques in the AEC domain

Technique (Ref.)	Detected primitive or object	Limitation
Connected Component Labelling, Hough transform and OCR with a SVM classifier (Maity *et al*. 2017)	• Detected the callouts based on circle detection. • Sheet number extracted. • Localized the text in the document sheets.	• Limited to detect the callouts and the text within it.
Collinearity test, component labelling and distance transform (De *et al*. 2014)	• Detected the line, circle and arcs in engineering drawings. • Classified the line types. • Arrowhead detected.	• Limited to detect the basic primitives such as lines, arcs and circles.
Deformable template matching (Valveny and Martí 1999)	• Recognized the symbols represented in straight lines.	• Limited to recognize lineal symbols. • Complexity of the algorithm is high due to the collection of set of deformations (translations, rotations and scaling).
Morphological binary erosion and dilation and Connected component analysis (Ahmed *et al*. 2011)	• Detected the walls in the architectural floor plan prior to text/graphics segmentation	• High recall value was achieved for the purpose of text/graphics segmentation but certain thresholds were empirically obtained and applied which means it is not scale invariant.
Network of constraints (Ah-Soon and Tombre 2001)	• Detected symbols represented in thin lines such as windows and doors.	• Errors due to noise and the approximation. • Number of features in the network can get large. This increases computing time and memory requirement due to increased searching process. • Network relies on the quality of the vectorization.

each constraint and then merges the features to get the symbol. Although this single step look-up method could be a strong feature of the approach, the drawback is that it can be complex when looking for a single symbol or subset of symbols.

As reviewed in this section, previous symbol recognition techniques that were used in the AEC domain mostly focuses on processing the raster graphics despite that 2D drawings nowadays are produced in vector-based files. The following section further reviewed the previously used symbol recognition techniques in the AEC domain with the focus on reconstructing the building model.

2.3 Utilized symbol recognition techniques in previous studies in the AEC domain for the purpose of building model reconstruction

In this section, the authors further compared the types of symbol techniques that were used in previous studies which aimed to reconstruct the building information model in the AEC domain (Table 3).

Components in the architectural drawings and structural drawings usually have a spatial, topological, or semantic relationship among them. Therefore, a number of papers utilized the graph based methods built upon domain knowledge (Gimenez *et al*. 2016, Lewis and Séquin 1998). Moreover, loop finding (i.e., cycle searching) approach to obtain space information and the relationship between spaces was also one of the commonly used techniques (Lu *et al*. 2005, Zhu *et al*. 2014). These methods are applicable to architectural and structural drawings mainly because the components in the drawings are usually connected to each other. For example, door and window symbols are connected to walls, spaces are connected with walls between them and structural components are usually placed along the horizontal and vertical grid lines. On the other

Table 3 Advantage and limitation of previous symbol recognition techniques for building model reconstruction in the AEC domain

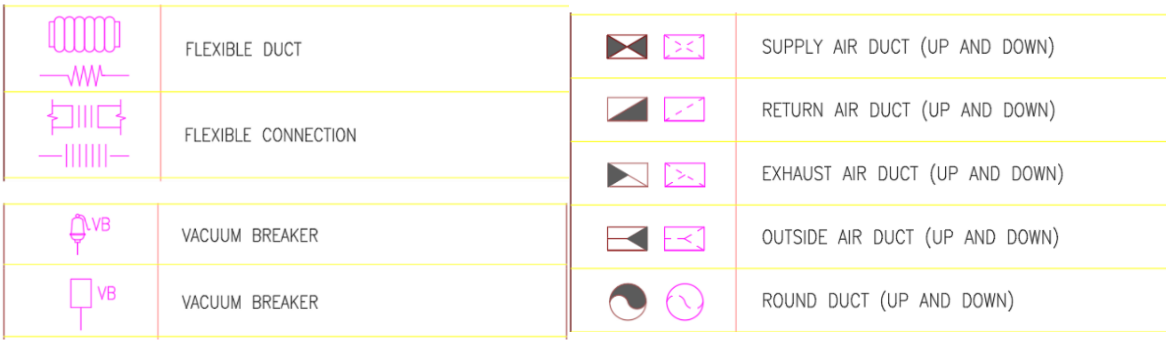
Technique (Ref.)	Advantage	Limitation
Graph matching and region growing (Gimenez *et al*. 2016)	By using both geometric elements and text enables to represent the relationships of the components in spaces.	• Walls are assumed to be all straight. Thus, curved walls are not considered. • Region growing method to recognize spaces relies on the text elements. Some drawings might not have text.
k-nearest neighbors algorithm (k=1) (Bortoluzzi *et al*. 2017)	XY coordinates can simply be compared with the room boundaries to identify rooms and assign numbers to each room.	• The boundary information of each room needs to be defined using a Revit API prior to room numbering. • Room tags outside the room (e.g., small closet) cannot be automatically identified.
Line-growing technique (Xu *et al*. 2015)	Suitable for simply recognizing rooms.	• Extracting rooms based on Minimum Bounding Rectangle (MBR) is only applicable for rectangular shaped spaces.
Shape-opening graph (Zhu *et al*. 2014)	Once the graph is built it enables fast loop extruding for reconstruction.	• Not able to recognize isolated walls which are not adjacent to any component.
Semantic constraints (Lu *et al*. 2005)	Works for drawings that has imperfections (e.g., disjoint errors).	• It cannot recognize the graphical sections of structural objects in the form of table since it is a top-down model based on semantic relation. • Not able to address the object representation problem (e.g., multiple symbols indicating steel).
Network of constraints (Dosch *et al*. 2000)	The identical algorithm used for recognition can also be used when building the network.	• Approximation of curves by polylines still causes error during symbol recognition. • Certain threshold is given when measuring error between the searched symbol and the candidate features.
Graph matching (Lewis and Séquin 1998)	Adjacency information can be developed which is useful for editing.	• Requires clean input data due to errors (e.g. disjoints).

hand, in mechanical drawings, such methods can be applied to recognize the connected duct components but have a limitation when recognizing other symbols that contain complex geometric features. The details of the advantage and limitation of such techniques are described in Table 3.

Graph matching approach is usually computationally complex in terms of running time. So, it is applicable to small sized graphs. Moreover, a significant amount of effort is required to clean the data in the pre-processing step in order to use the graph-based approach on vector-based drawings. Also, it was noticed that it is not applicable to recognize symbols that are isolated and not connected to other components. In the following section, the authors described the characteristics of the 2D building mechanical drawings and the challenges that need to be addressed.

2.4 Characteristics of 2D building mechanical drawings

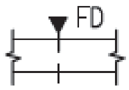
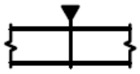
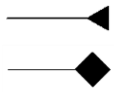
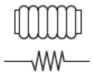
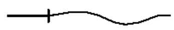
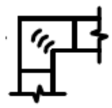
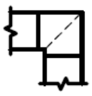
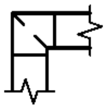
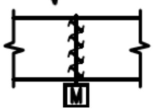
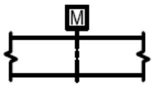
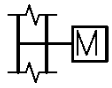
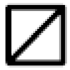
A common factor that makes applying symbol recognition techniques on AEC domain drawings difficult is because variation exists among different drawings. Although the United States National CAD Standard (National Institute of Building Sciences building SMART alliance 2014) exists, there are some symbols where the width is user-defined. In addition, the possibility of drafting errors cannot be simply ignored (Cho and Liu 2017). On top of that, the 2D building mechanical drawing has several different characteristics compared to architectural or structural drawings which make symbol recognition more challenging.



(a) Multiple symbols indicating identical objects (b) Symbols with similar shapes

Fig. 1 Examples of symbols in the symbol description document (from Cho *et al*. 2018b)

Table 4 Symbol variations among different drawing sets

Symbol descriptions of mechanical components by building	Building A (Gates Hillman)	Building B (Warner Hall)	Building C (Hamburg Hall)
Fire damper (vertical or horizontal)			
Flexible duct			
Mitered elbow with turning vanes			
Motorized damper			
Return air component			

First of all, the contour-based method that only extracts features from external contour would not address the recognition problem because a number of different symbols exist with similar rectangle shape contours. Utilizing a method that could reflect the features of the internal region shape contents is important. A recursive searching method that could distinguish the small details regarding the structural relationship is inevitable to utilize due to very similar symbols (Fig. 1).

Second, many arrow annotations and texts are positioned around the mechanical components in the drawing. An algorithm that searches the surrounding region of the symbol is required to retrieve the detailed information of the equipment such as equipment type, ID number and airflow in cubic feet per minute (CFM) from the connected annotation and equipment reference symbols nearby.

Lastly, multiple symbols indicating an identical object exists. For example, there could be two ways of drawing a flexible duct (Fig. 1(a)). The symbol recognition method would need to generate different feature library for these cases. In addition, a variation of symbols exists among different drawing sets (Table 4). Thus, the variation of symbols also needs to be considered in order to develop a generalized framework that could reconstruct the building information model from different drawing sets.

Furthermore, frequently occurring components are usually drawn with elements so-called ‘Block Reference’ for convenience in AutoCAD. The authors investigated 55 drawings from 5 different buildings. The drawings from three buildings that were built in the 1900s were not drawn using block references. But in recent two building drawings that were produced in 2008 and 2013, 91.7% out of the total number of mechanical components were drawn using block reference. Relatively old drawings used fewer block references but since the resizable block (i.e., Dynamic Block) feature was added to AutoCAD in 2005, block references became actively used. Therefore, the authors focused on the recent drawing trend and studied how to recognize symbols in block references in this study.

3. Research method

3.1 Framework

Based on the research of previous studies and the characteristics of 2D building mechanical drawings that were described in section 2, the authors proposed a framework for recognizing symbols in the mechanical drawings (Fig. 2). The method is based on a hybrid approach using vectorial signatures proposed in previous studies (Dosch and Lladós 2003, Schettini 1996, Zhang and Liu 2007).

The vectorial signature matching approach is known to be invariant to rotation, scale and other affine transformations (Valveny and Dosch 2003). In addition, the vectorial signatures method with a primitive-pair relationship can recognize symbols with partial information since it calculates the number of common relationships (Zhang and Liu 2007).

The proposed framework utilizes two types of documents as input: the 2D building mechanical drawings and the symbol description document. The symbol description document is used to generate the library symbols which plays the role of a symbol database. And the symbols that appear in the 2D drawings which are stored as block reference elements are used for test symbols. The goal is to recognize and localize the test symbols by comparing them with the library symbols using the matching test.

The first step starts with pre-processing the inputs. This step includes reading in the two input documents into the system, symbol description table processing and cleaning the data regarding the symbols. Python libraries called ‘ezdxf’ and ‘dxfgrabber’ were used to read in the inputs. These libraries allow the system to directly work with the elements such as LINE, LWPOLYLINE, POLYLINE, ARC, CIRCLE, HATCH, SOLID, MTEXT, TEXT, INSERT and ATTDEF which are the elements that are used to generate the drawing documents in AutoCAD. Moreover, block reference objects which consist of a combination of these elements can also be read into the system separately. A block can also include another block, the so-called ‘nested block’. Thus, the blocks need to be iteratively searched and all the elements need to be pre-processed to basic geometric primitives for representation.

For instance, Fig. 3 shows one example of a decomposed symbol (supply diffuser) in AutoCAD. This symbol consists of three rectangular shapes that are drawn using the LWPOLYLINE element and two diagonal lines that are drawn as LINE element. Unless the symbol is decomposed as shown in Fig. 3, it is hard to know whether if the symbol is totally drawn

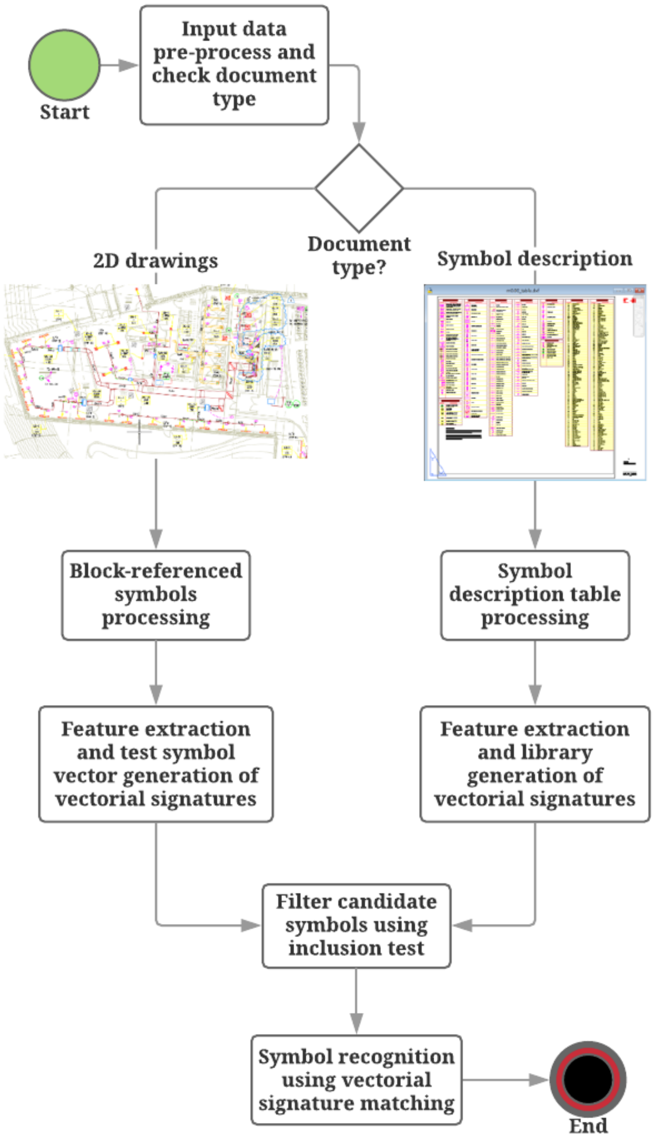


Fig. 2 Framework of the proposed study

with LINE elements or a mixture with LWPOLYLINE elements. Let's assume one is creating the 2D drawing based on the company's symbol description. Someone could decide to draw the same symbol with two triangles using the LWPOLYLINE element and have the two diagonal lines of the triangles overlap instead of drawing the boundary rectangle and the two lines shown in Fig. 3.

Without the process of data cleaning, this would lead to different representation which will eventually affect the successive symbol recognition process. Therefore, all LWPOLYLINE elements need to be separated to basic lines and all lines need to be decomposed based on their intersection points. The process of decomposing the lines is required because of the two diagonal lines shown in Fig. 3 can also be drawn with four lines depending on the designer.

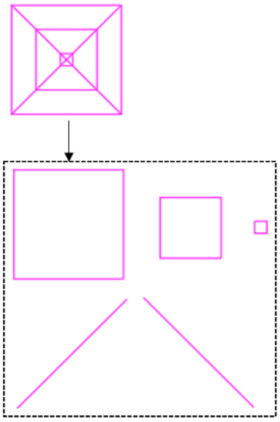
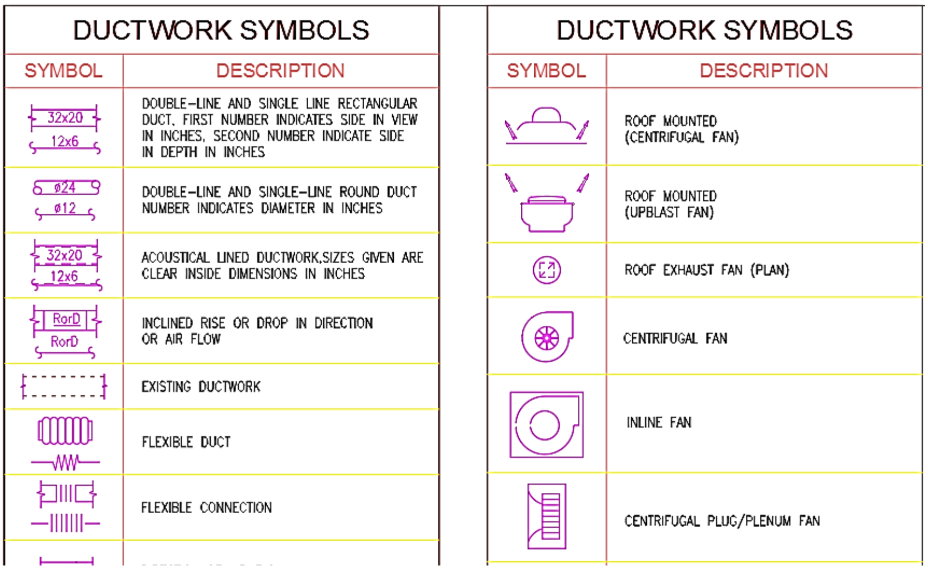


Fig. 3 Example of decomposed symbol (supply diffuser)



DUCTWORK SYMBOLS	
SYMBOL	DESCRIPTION
32x20 12x6	DOUBLE–LINE AND SINGLE LINE RECTANGULAR DUCT. FIRST NUMBER INDICATES SIDE IN VIEW IN INCHES. SECOND NUMBER INDICATE SIDE IN DEPTH IN INCHES
ø24 ø12	DOUBLE–LINE AND SINGLE–LINE ROUND DUCT NUMBER INDICATES DIAMETER IN INCHES
32x20 12x6	ACOUSTICAL LINED DUCTWORK.SIZES GIVEN ARE CLEAR INSIDE DIMENSIONS IN INCHES
RorD RorD	INCLINED RISE OR DROP IN DIRECTION OR AIR FLOW
	EXISTING DUCTWORK
	FLEXIBLE DUCT
	FLEXIBLE CONNECTION

DUCTWORK SYMBOLS	
SYMBOL	DESCRIPTION
	ROOF MOUNTED (CENTRIFUGAL FAN)
	ROOF MOUNTED (UPBLAST FAN)
	ROOF EXHAUST FAN (PLAN)
	CENTRIFUGAL FAN
	INLINE FAN
	CENTRIFUGAL PLUG/PLENUM FAN

Fig. 4 A portion of the symbol description document

Moreover, there are cases where unnecessary lines exist which are overlapping in the symbol due to human error. In addition, overlapping lines also occurs from the previous LWPOLYLINE decomposition process too. Thus, these overlapping lines which can be considered as noise need to be eliminated. The elimination of overlapping lines not only prevents generating unwanted features in the successive step but also reduces the amount of data that the system needs to handle.

To pre-process the symbol description document, the authors implemented a table processing step for a generalized approach. Every drawing set contains a symbol description document and is in the form of a table (Fig. 4). The symbol description document usually consists of 5~7 columns of symbol and abbreviation description. However, this table is drawn with the LINE element. Therefore, the authors developed a table processing module that recognizes each cell in the table based on the vertical and horizontal lines of the table. Each cell with a symbol in it is assigned with an ID number and the text description on the right cell of the symbol cell is stored to be used as labels in the recognition phase.

Then, the elements in the symbol cells are also pre-processed to basic geometric primitives. By using the extracted primitives features of symbols from the symbol description table, a library of the vectorial signatures of each symbol is generated. A set of primitive-pair relationship features of the symbols are each stored as a signature. The description of the utilized features is explained in more detail in section 3.2.

The following step after the generation of the library of vectorial signatures is filtering the candidate symbols for recognition. There is two unique information that could be used as a filter: text information and information from entities such as SOLID and HATCH elements. The authors used the text, element and the junction feature information as filters for inclusion test.

For the symbol recognition phase, the authors utilized a vectorial signature matching method. In this paper, the authors focused on recognizing the symbols that were drawn as block references. For each block reference symbols, the number of common relationships is calculated with each vectorial signatures in the library and a description label is assigned based on similarity calculation. As a result, a map of recognized and localized symbols is generated as an output at the end of the framework.

3.2 Recognizing symbols in 2D building mechanical drawings

3.2.1 Symbol recognition using the angle feature

The authors have assumed the symbols in 2D drawings would have been drawn as how it is described in the symbol description document.

However, multiple inconsistency issues were found. One example is shown in Fig. 5. In the symbol description document, the supply diffuser symbol has two text information and a line between the texts but in the actual symbol in the 2D drawing, there is no text information or any

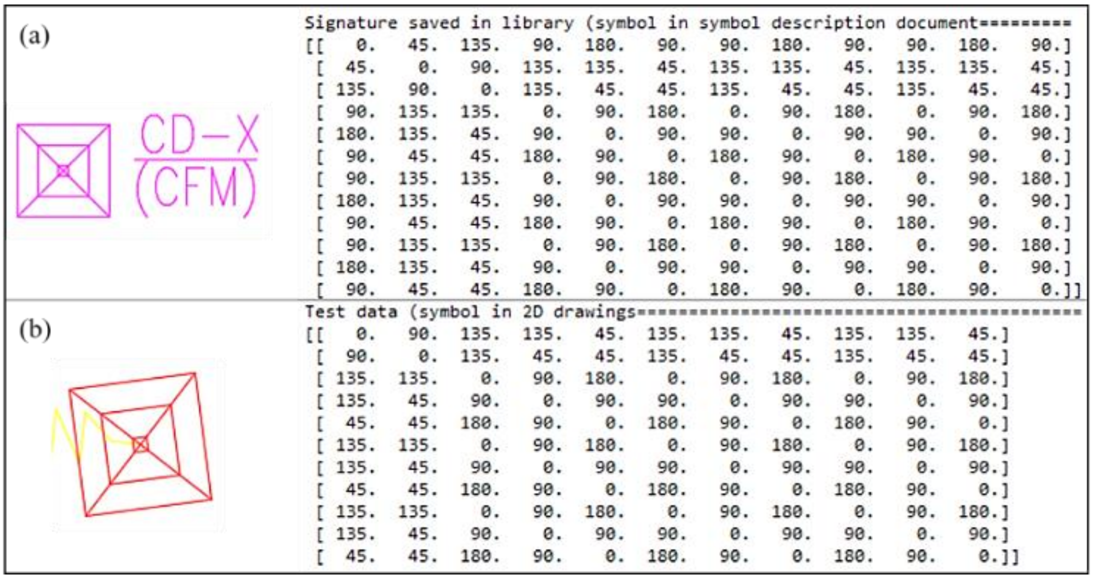


Fig. 5 Vectorial signature of the diffuser symbol (a) signature of the symbol in the symbol description document, (b) signature of the symbol in the 2D drawing (from Cho *et al*. 2018b)

Table 5 Symbol similarity among different symbols of mechanical components

Symbol similarity (test / library)	Supply diffuser	Return register	VAV box	Linear diffuser
Supply diffuser	**0.752**	0.689	0.642	0.583
Return register	0.695	0.680	**0.704**	0.629
VAV box	0.306	0.311	**0.429**	0.381
Linear diffuser	0.591	0.620	**1.000**	0.889

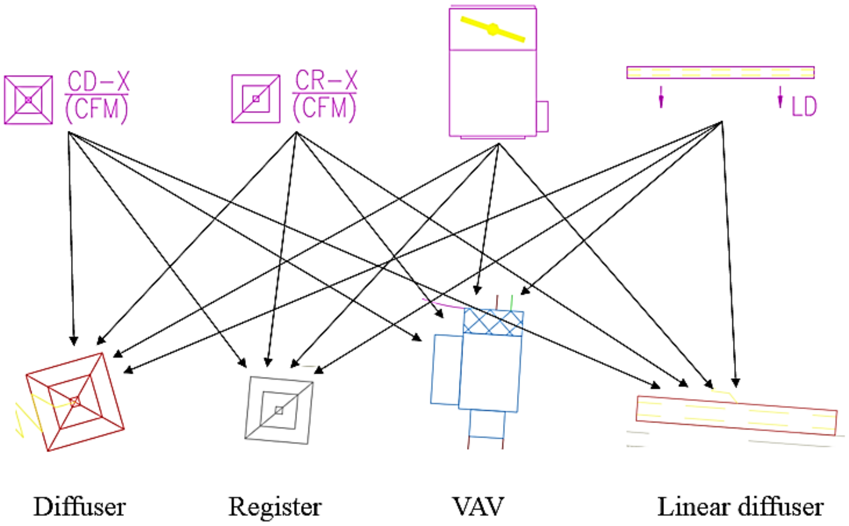


Fig. 6 The four symbols used for symbol recognition test in experiment 1 and 2

additional line. This becomes an issue because it leads to non-identical size (i.e., shape) of the vectorial signatures which needs to be compared.

The vectorial signature of the LINE and LWPOLYLINE element in the supply diffuser symbol were generated based on the angles between lines which was introduced in Zhang and Liu (2007). Each number in the array indicates the angle in degrees between the unit vectors of lines. But due to the inconsistent number of line elements, the size of the signature can be different. Thus, the Jaccard similarity method was utilized to measure the similarity between the two signatures. This method searches the intersection and union which is applicable for measuring the similarity of different size of samples. Each row in the signature that is stored in the library was compared with every row in the signature of the test data (i.e., 2D drawing). In this case (Fig. 5), the Jaccard similarity was calculated a 12x11=132 number of times and the total sum of the similarity measurements was divided by the number of measurement which is 132. As a result, a number in the range [0, 1] is obtained, 0.752 (Table 5).

The authors have conducted the symbol similarity test with four types of mechanical components that are drawn with LINE and LWPOLYLINE elements (Fig. 6). The higher the number in the range [0, 1] means that it is likely a match. The maximum value of each similarity test is described in bold text in Table 5. However, due to inconsistency between the symbols in the symbol description document and the 2D drawings, the result in Table 5 shows that components such as return register and linear diffuser did not match properly.

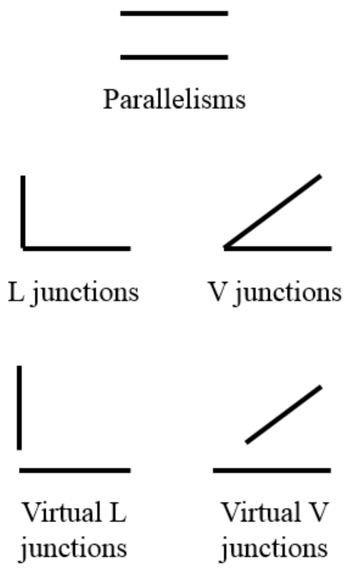


Fig. 7 The five relation features

3.2.2 Symbol recognition using multiple relation features

The test result in Table 5 has shown that solely using the vectorial signature of angles is not sufficient and that needs additional complementary approaches to achieve higher symbol recognition rate. Thus, the author further conducted a second test using the vectorial signature of multiple relations that was suggested in Dosch and Lladós (2003). The symbol elements were transformed into a vector that includes the following five relation features: the number of parallelisms, L junctions, V junctions, virtual line L junctions and virtual line V junctions (Fig. 7).

In the second test, the number of each features shown in Fig. 7 were counted instead of the angle feature that was used in the previous test.

The symbol recognition phase of the second test consists of two steps. First, the algorithm searches for the candidate symbols using the inclusion test (Dosch and Lladós 2003). It finds the candidate symbols which has the corresponding relation features. The second step calculates the quality of the match (Schettini 1996) between the symbol of interest and the candidate symbols. The form of the calculation is as the following

$$\mathrm{D}(KS_i, CS_j) = \sum_{k=1}^{K} |f_{ik} - f_{jk}| w_k \tag{1}$$

where K is the total number of features which in this case is 5, f_{ik} is the value of feature k of the known symbol(KS) which are the features extracted from the symbols from the symbol description document, f_{jk} is the value of the feature k of the candidate symbol(CS) which are the features extracted from the symbols from the actual 2D drawings and w_k is the weighting of each feature or normalization factor. The representation and calculation details are shown in Fig. 8.

The symbols that obtain the minimum value of the sum of feature differences multiplied by each feature weights are considered a match (Fig. 8). The result of the matching of the second test is shown in Table 6 and two test examples of hypothesis test are described as the following (Fig. 9).

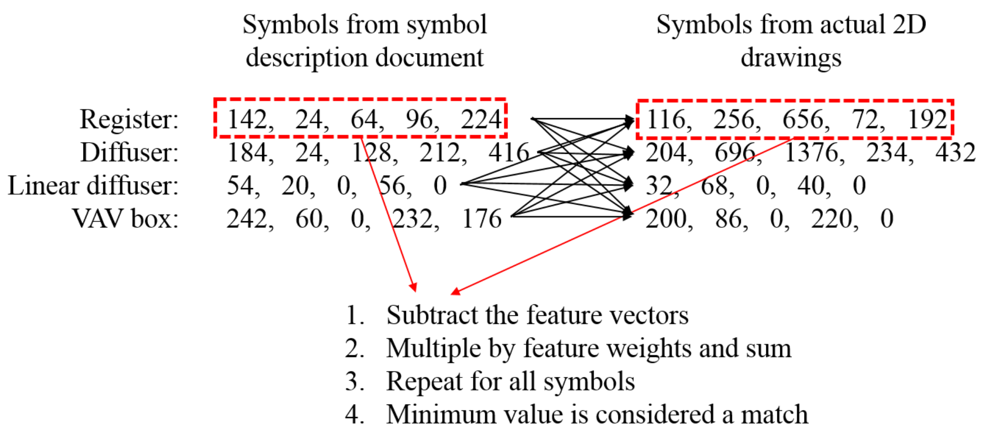


Fig. 8 The process of calculating the quality of the match between symbols

Table 6 Symbol recognition based on the quality of the match

Quality of the match (test / library)	Supply diffuser	Return register	VAV box	Linear diffuser
Supply diffuser	**10.2**	14.7	18.5	19.5
Return register	14.3	**9.2**	11.2	13.8
VAV box	52.8	52.8	**22.4**	39.2
Linear diffuser	41.6	38.4	18.4	**6.4**

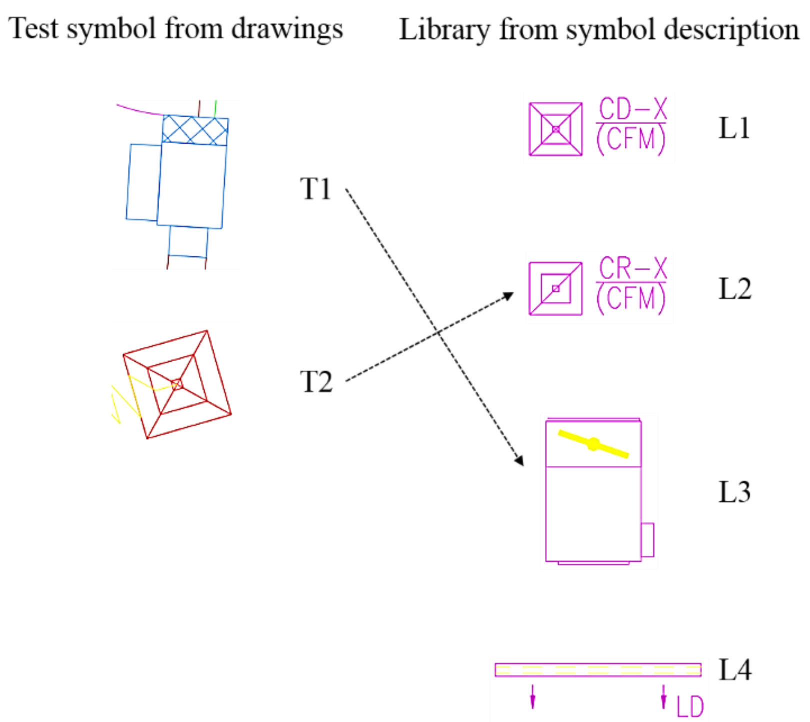


Fig. 9 Hypothesis test

Example: test symbol T1 (VAV box)
L1 → Match (L1, T1) is 52.8
L2 → Match (L2, T1) is 52.8
L3 → Match (L3, T1) is 22.4
L4 → Match (L4, T1) is 39.2
Classification: minimum Match (Li, T1) → T1 is L3

Example: test symbol T2 (Return register)
L1 → Match (L1, T2) is 14.3
L2 → Match (L2, T2) is 9.2
L3 → Match (L3, T2) is 11.2
L4 → Match (L4, T2) is 13.8
Classification: minimum Match (Li, T2) → T2 is L2

The minimum value of each quality of match test is illustrated in bold text in Table 6. The method used in experiment 2 has proved successful since it achieved a 100% recognition rate when matching the four test symbols from the 2D building mechanical drawings with the library symbols generated from the symbol description document.

3.2.3 Symbol recognition using multiple relation features and additional relationship features

In addition, the author conducted a third symbol recognition experiment on six drawings with twelve additional relationship features (i.e., 5+12 = 17 in total) and newly added rules to test on more symbols (Fig. 10).

In this experiment, a symbol description document was manually generated for testing purpose. 24 different type of symbols that actually appeared in the six drawings were considered and the corresponding symbols were copied from the original symbol description document to be used as input (Fig. 11). And the twelve features that were added include the following: the number of disjointness, intersection and tangency between lines and circles, between arcs and circles, between circles and between arcs and lines (Zhang and Liu 2007).

Rules were established to use the information of TEXT, SOLID and HATCH elements in the symbols to distinguish symbols that have an identical vectorial signature and also to use as filters to generate a list of candidate symbols with higher match possibility. The reason why having rules

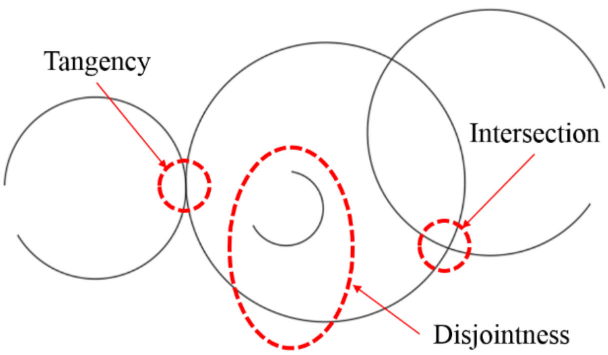


Fig. 10 The additional relationship features

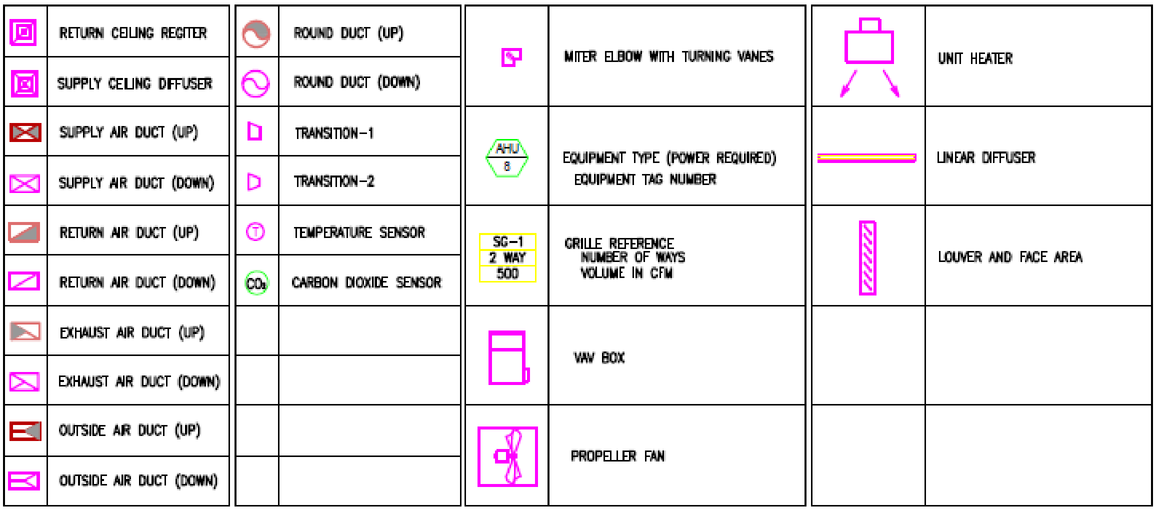


Fig. 11 Twenty-four type of symbols used for symbol recognition test in experiment 3

Table 7 The recognition rate result of symbol recognition test on six different drawings

Drawings (D1~D6)	D1	D2	D3	D4	D5	D6
# of correct match	10	8	13	12	11	12
# of type of symbols	14	12	15	16	15	16
Recognition rate (%)	71%	67%	87%	75%	73%	75%

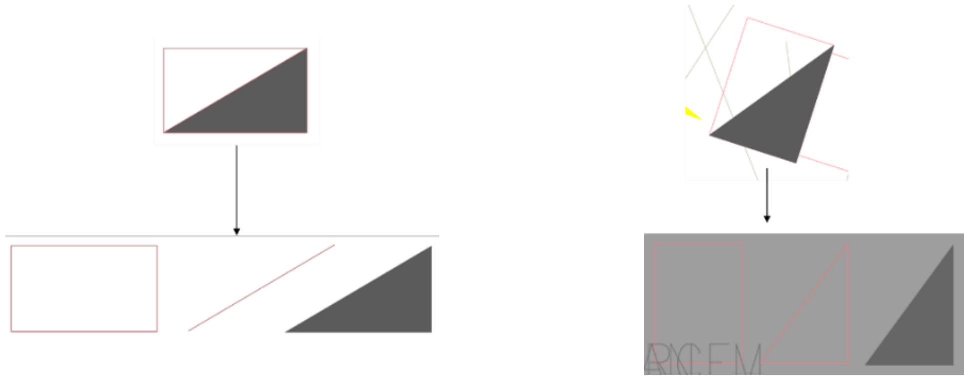
is necessary is because there are similar symbols that are distinguished with the SOLID or HATCH element. For example, the direction of the air flow can be distinguished between the ‘RETURN AIR DUCT (UP)’ and ‘RETURN AIR DUCT (DOWN)’ symbols by using the SOLID element that is colored in grey (Fig. 11).

As a result, recognition rates of 71%, 67%, 87%, 75%, 73% and 75% were obtained when tested on six different drawings from the same building (Table 7). The major reason for mismatch cases was due to the inconsistency of symbols between the symbol description document and the symbols in the 2D drawings.

Despite the fact that a number of pre-processing steps such as the elimination of overlapping lines and decomposition of lines were applied, the result has shown that additional process of error fixing is required to resolve the variation issue. Although 100% accuracy was not obtained in this study when tested with 24 different symbols, the experimental results are satisfying because the reason of mismatch cases was not mainly due to the proposed framework but rather because several symbols in 2D drawings were not drawn as it was shown in the symbol description document.

4. Conclusions

The authors believe that with the rapid and low-cost approach of generating BIM would lower the barrier of adopting BIM for FM. And symbol recognition is one of the essential steps to obtain



(a) Return air duct symbol from the symbol description document

(b) Return air duct symbol from the actual 2D drawing

Fig. 12 Inconsistency issue between symbols

such an approach. Thus, the goal of this paper was to study previously used symbol recognition techniques and propose a framework that could automatically recognize symbols in the 2D building mechanical drawings for 3D reconstruction purpose.

Previously used symbol recognition techniques were reviewed by three symbol recognition phases and the non-relevant techniques that are not applicable to this study were initially filtered out. Furthermore, based on the research on previous studies and considering the unique characteristics of symbols in the building mechanical drawings, the authors developed a framework that would recognize and localize the mechanical components from the mechanical drawings. The main contribution of this paper is that it analyzed the unique characteristics of 2D building mechanical drawings and the block references regarding the challenges of recognizing the symbols of building mechanical components. As a result of the proposed framework, the authors achieved a symbol recognition rate of 75% on average when tested on six different drawings.

However, several limitations of the suggested method were discovered. One major factor that reduced the accuracy in the test was due to inconsistency (Fig. 12). The library symbol (Fig. 12(a)) of the return air duct symbol looks identical with the test symbol (Fig. 12(b)) from the drawing in the first glance. But when the authors decomposed the two symbols in AutoCAD, the vectorial signatures of these two symbols turned out to be different because the test symbol was not drawn as it is described in the symbol description document. The test symbol (Fig. 12(b)) had a triangle shape in it while the library symbol (Fig. 12(a)) used a line element to form the symbol. As a matter of fact, the test symbol (Fig. 12(b)) is drawn in a wrong way because the diagonal was supposed to be drawn from the bottom left corner to the top right corner as described in the library symbol (Fig. 12(a)). It is obviously a human error case but this also could be considered as a variation issue since the test symbol does not indicate any other object.

Although this inconsistency and variation issue was caused by human error and was the major factor that lowered the accuracy of the symbol recognition rate, the authors are considering adding a refinement process by developing a reasoning mechanism that would exploit the knowledge of the air flow of the mechanical system as a potential solution. This approach would require an additional effort of integration of the recognized duct information from the previous study (Cho and Liu 2017) but it would be able to notice the human error and eventually improve the symbol recognition rate.

Another limitation with the current framework is the fact that the authors had manually generated the symbol description document in experiment 3 (Fig. 11). This was because multiple symbols were included in the symbol description table and needed to be separated to generate the library symbols. However, an automatic symbol separation mechanism during the symbol description table processing would be required to achieve a more generalized approach.

Furthermore, the issue regarding the multiple symbols indicating identical symbols that was mentioned in section 2.4 (Fig. 1) also needs to be addressed to improve the symbol recognition coverage of building mechanical components. Therefore, the future study would aim to address the remaining challenges and explore more experiments using different building mechanical drawings. The authors may explore more sophisticated symbol recognition methods and optimized feature matching algorithm based on the result of future studies. Moreover, using the symbol recognition result, the authors will further focus on studies on how to reconstruct BIM based on the mechanical objects that were recognized.

Acknowledgments

The research described in this paper was supported by the Department of Civil and Environmental Engineering at Carnegie Mellon University.

References

Ah-Soon, C. and Tombre, K. (2001), "Architectural symbol recognition using a network of constraints", *Pattern Recog. Lett.*, **22**(2), 231-248.

Ahmad, N. and Haque, A.A. (2001), "Manufacturing feature recognition of parts using DXF files", *4th International Conference on Mechanical Engineering*, **6**, 111-115, Istanbul, December.

Ahmed, S., Weber, M., Liwicki, M. and Dengel, A. (2011), "Text/graphics segmentation in architectural floor plans", *Document Analysis and Recognition (ICDAR)*, *2011 International Conference*, 734-738, IEEE, Peking, China, September.

American Institute of Architects (1997), "AIA Document B141 - 1997 Part 1: Standard form of agreement between owner and architect with standard form of architect's services", U.S. Securities and Exchange Commission, Washington, D.C, U.S.A. https://www.sec.gov/Archives/edgar/data/1124105/000119312507061689/dex1021.htm

Belongie, S., Malik, J. and Puzicha, J. (2001), "Matching shapes", *ICCV 2001, Proceedings of Eighth IEEE International Conference*, 454-461, IEEE, Vancouver, July.

Borgefors, G. (1984), "Distance transformations in arbitrary dimensions", *Comput. Vis. Graphic Image Process.*, **27**(3), 321-345.

Bortoluzzi, B., Sobieraj. D. and McArthur, J. (2017), "Automating creation of facility and energy management Building Information Models", *Lean and Computation in Construction Congress,* Heraklion, Greece, July, 153-160.

Cho, C.Y. and Liu, X. (2017), "An automated reconstruction approach of mechanical systems in building information modeling (BIM) using 2D drawings", *Comput. Civil Eng.*, 236-244, Seattle, Washington, U.S.A., June.

Cho, C.Y. Liu, X. and Akinci, B. (2018b), "Recognizing symbols in building mechanical drawings for automated building information models reconstruction", *17th International Conference on Computing in Civil and Building Engineering*, 295-302, Tampere, Finland, June.

Cho, C.Y., Liu, X. and Akinci, B. (2018a), "Automated building information models reconstruction using 2D mechanical drawings", *35th CIB W78 2018 Conference (Advances in Informatics and Computing in Civil* *and Construction Engineering)*, 505-512, Chicago, Illinois, U.S.A., October.

Cordella, L.P. and Vento, M. (1999), "Symbol and shape recognition", *International Workshop on Graphics Recognition*, 167-182, Springer, Berlin, Heidelberg, Germany, September.

De, P., Mandal, S., Das, A. and Bhowmick P. (2014), "A new approach to detect and classify graphic primitives in engineering drawings", *2014 Fourth International Conference of Emerging Applications of Information Technology*, Kolkata, India, 243-248.

Dosch, P. and Lladós, J. (2003), "Vectorial signatures for symbol discrimination", *International Workshop on Graphics Recognition*, 154-165, Springer, Berlin, Heidelberg, Germany, July.

Dosch, P., Tombre, K., Ah-Soon, C. and Masini, G. (2000), "A complete system for the analysis of architectural drawings", *J. Document Anal. Recog.*, **3**(2), 102-116.

Fallon, K.K. and Palmer, M.E. (2007), "General buildings information handover guide: Principles", NISTIR 7417; National Institute of Standards and Technology, Gaithersburg, MD, U.S.A.

Gimenez, L., Robert, S., Suard, F. and Zreik, K. (2016), "Automatic reconstruction of 3D building models from scanned 2D floor plans", *Automat. Construct.*, **63**, 48-56.

Goshtasby, A. (1985), "Description and discrimination of planar shapes using shape matrices", *IEEE T. Pattern Anal. Machine Intell.*, **6**, 738-743.

Ha, D. and Eck, D. (2017), "A neural representation of sketch drawings", *arXiv preprint arXiv*:1704.03477.

Jain, A.K., Duin, R.P.W. and Mao, J. (2000), "Statistical pattern recognition: A review", *IEEE T. Pattern Anal.*, **22**(1), 4-37.

Lewis, R. and Séquin, C. (1998), "Generation of 3D building models from 2D architectural plans", *Comput. Aided Design*, **30**(10), 765-779.

Lin, X., Shimotsuji, S., Minoh, M. and Sakai, T. (1985), "Efficient diagram understanding with characteristic pattern detection", *Comput. Vis. Graphic Image Process.*, **30**(1), 84-106.

Liu, C., Wu, J., Kohli, P. and Furukawa, Y. (2017), "Raster-to-Vector: Revisiting floorplan transformation", *International Conference on Computer Vision (ICCV 2017)*, 2195-2203, Venice, Italy, October.

Lladós, J., Martí, E. and Villanueva, J.J. (2001), "Symbol recognition by error-tolerant subgraph matching between region adjacency graphs", *IEEE T. Pattern Anal. Machine Intelligence*, **23**(10), 1137-1143.

Loncaric, S. (1998), "A survey of shape analysis techniques", *Pattern Recog.*, **31**(8), 983-1001.

Lu, Q. and Lee, S. (2017), "A semi-automatic approach to detect structural components from CAD drawings for constructing as-is BIM objects", *Computing in Civil Engineering 2017*, 84-91, Seattle, Washington, U.S.A., June.

Lu, T., Tai, C.L., Su, F. and Cai, S. (2005), "A new recognition model for electronic architectural drawings", *Comput. Aided Design*, **37**(10), 1053-1069.

Maity, S.K., Seraogi, B., Das, S., Banerjee, P., Majumder, H., Mukkamala, S., Roy, R. and Chaudhuri, B.B. (2017), "An approach for detecting circular callouts in architectural, engineering and constructional drawing documents", *International Workshop on Graphics Recognition*, 17-29, Kyoto, November.

McArthur, J.J. (2015), "A building information management (BIM) framework and supporting case study for existing building operations, maintenance and sustainability", *Procedia Eng.*, **118**, 1104-1111.

Messmer, B.T. and Bunke, H. (1995), "Automatic learning and recognition of graphical symbols in engineering drawings", *International Workshop on Graphics Recognition*, 123-134, Berlin, Heidelberg, August.

National Institute of Building Sciences buildingSMART alliance (2014), "United States National CAD Standard®–V6: Module 6 – Symbols", *United States National CAD Standard*, U.S.A. https://www.nationalcadstandard.org/ncs6/pdfs/ncs6_uds6.pdf.

Santos, D.S., Dionísio, M., Rodrigues, N., Pereira, A. and Leiria, I.I.I. (2011), "Efficient creation of 3D models from buildings' floor plans", *J. Interactive Worlds*, **2011**(2011), 1-30.

Schettini, R. (1996), "A general-purpose procedure for complex graphic symbol recognition", *Cybernet. Syst.*, **27**(4), 353-365.

Tabbone, S. and Terrades, O.R. (2014), "An overview of symbol recognition", *Handbook of Document Image Processing and Recognition*, 523-551, Springer, London, United Kingdom.

Tabbone, S., Wendling, L. and Salmon, J.P. (2006), "A new shape descriptor defined on the Radon

transform", *Comput. Vis. Image Und.,* **102**(1), 42-51.
Valveny, E. and Dosch, P. (2003), "Symbol recognition contest: A synthesis", *International Workshop on Graphics Recognition*, 368-385, Berlin, Heidelberg, Germany, July.
Valveny, E. and Martí, E. (1999), "Application of deformable template matching to symbol recognition in handwritten architectural drawings", *Document Analysis and Recognition, ICDAR'99: Proceedings of the Fifth International Conference*, 483-486, IEEE, Bangalore, India, September.
Ventura, A.D. and Schettini, R. (1994), "Graphic symbol recognition using a signature technique", *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, 533-535, IEEE, Jerusalem, Israel, October.
Xu, D., Jin, P., Zhang, X., Du, J. and Yue, L. (2015), "Extracting indoor spatial objects from CAD models: A database approach", *International Conference on Database Systems for Advanced Applications*, 273-279, Hanoi, Vietnam, April.
Yin, Z., Guan, Y., Chen, S., Wu, W. and Zhang, H. (2013), "Off-line programming of robotic system based on dxf files of 3d models", *IEEE International Conference on Information and Automation (ICIA 2013)*, 1296-1301, Yinchuan, China, August.
Zhang, D. and Lu, G. (2004), "Review of shape representation and description techniques", *Pattern Recog.*, **37**(1), 1-19.
Zhang, W. and Liu, W. (2007), "A new vectorial signature for quick symbol indexing, filtering and recognition", *9th International Conference on Document Analysis and Recognition (ICDAR 2007)*, 536-540, Curitiba, Paraná, Brazil, September.
Zhi, G.S., Lo, S.M. and Fang, Z. (2003), "A graph-based algorithm for extracting units and loops from architectural floor plans for a building evacuation model", *Comput. Aid Design*, **35**(1), 1-14.
Zhu, J., Zhang, H. and Wen, Y. (2014), "A new reconstruction method for 3D buildings from 2D vector floor plan", *Comput. Aid Design Appl.*, **11**(6), 704-714.

CC