A new method to recognize Dimension Sets and its application in Architectural Drawings

Yalin Wang, Long Tang, Zesheng Tang
P O Box 84-187, Tsinghua University Postoffice
Beijing 100084, P.R.China.   Email: wyl@sun5.thit.tsinghua.edu.cn

Abstract - Recognition of dimension sets is a prerequisite for engineering drawings interpretation. A new heuristic algorithm for dimension sets recognition is presented. Following the recognized text strings, the algorithm recognizes the dimension lines and other components of the dimension set according to the dimension set syntaxes. The advantages of this algorithm to the architectural drawings are unmatched. A data structure of planar position index is used and evaluated. Experimental results are also presented

Key word: Dimension Set Recognition, Engineering Drawings Interpretation, Text Segmentation, Pattern Recognition, architectural drawings

I. Introduction

Engineering drawings are important vehicles to convey the mechanical and manufacturing information for 3-D solid objects. Traditionally, the drawings are manually prepared on the drawing board using paper-and-pencil drafting techniques. With the development of computer-aided design (CAD) and computer-aided manufacturing (CAM) techniques, more and more engineering drawings are in CAD formats. There are many merits that can demonstrate the conversion[1]: easy in storage, retrieval, and query, convenient in revision, and cheap in filing, copying, accessing, and preparing these drawings.

Many of the existing paper engineering drawings are still active and operational, and there is a demand for a fast and reliable means to transform them into some standard CAD/CAM format. Manually transferring paper drawings into such a format is labor-intensive and time-consuming, so it has become extremely desirable for the computer to process a wide variety of paper drawings and automatically create the CAD format drawings. Automatic engineering drawings input and recognition system can serve for this function.

Such systems should recognize not only the geometry primitives which represent the object contours but also the annotation primitives which give the measurement of the object. Dimension sets, which describe the logical relations among geometry primitives and annotations, should also be correctly recognized and interpreted, as it understanding is a prerequisite for the higher level understanding of engineering drawings.

Some approaches on recognizing dimension sets in the engineering drawings have been achieved[2,3,4,5]. Dori[2]presented the fundamental method for recognition of engineering drawings which starts with arrowhead recognition, including the definition of dimension components with a set $\Omega = \{C, W, S, L, A, T\}$, the concept of symmetry and regularity of dimensions on a functional basis. Min et al. [3] presented a web grammar based on arrowhead-match in accordance with the PeopleÆs Republic of China national standard GB4458-84 for machine engineering drawings. Their method can recognize 27 dimension patterns and 48 subpatterns which are based on morphological and functional classifications. Although their method has more practical implementation background, they assume that preprocessing work can supply the method with all the identified arrowheads and texts. However, due to the complexity of real-life engineering drawings and the presence of noise, it is still a tough problem for the vectorization module to supply a solid basis for their method.

Kasturi[4] developed a new rule-based text/graphics separation algorithm and a model-based procedure for detecting arrowheads in any orientation that are drawn to ANSI drafting standards. LangranaÆs method [5] recognize dimension sets based on morphological filters. With a manual input file of text, this method reconstructed geometry using variational geometry theory. Their methods all begin with the scanned image and get the recognized dimension sets finally. But the inherent difficulty with the arrowhead recognition makes them strongly dependent on the image quality. It also seems time-consuming to integrate the text blocks with the corresponding leaders and the feature control frames.

In this paper, we propose a new algorithm to recognize dimension sets in the architecture drawings (an example is shown in Fig. 1) based on dimension sets syntaxes, which is discussed in Section II. In Section III, a data structure named Planar Position Index(PPI) is briefly described, which is employed to fully realize the new algorithm. The advantages of this algorithm in building costs estimation on architecture drawings are unmatched, which is shown in Section IV. Finally, a summary and future work are given in Section V.

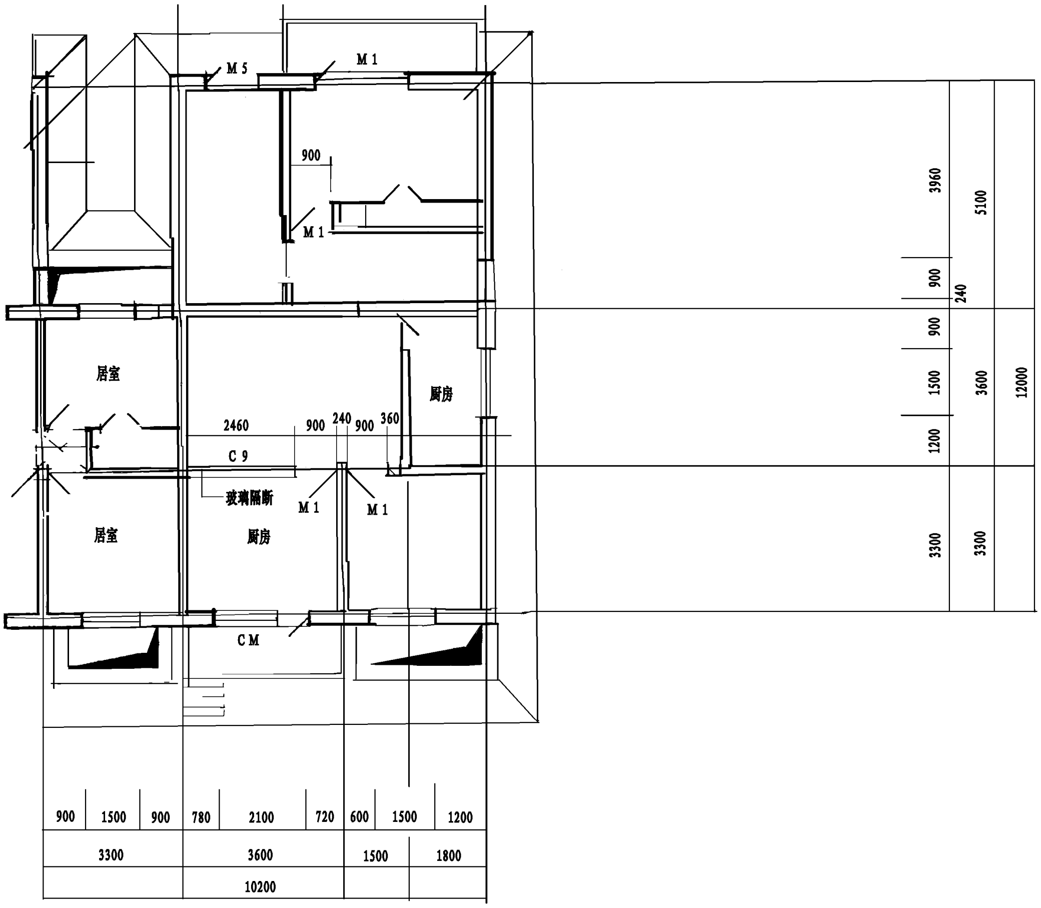


Fig. 1 An example of architectural drawing

II. Recognition of Dimension Sets in Architectural Drawings

Dimension sets are mainly used to define geometric characteristics such as lengths and relative locations of the various entities in the drawings. They consist of entities such as dimension lines, witness lines and dimensioning text. The block diagram of our dimension sets recognition procedure is shown in Fig. 2.

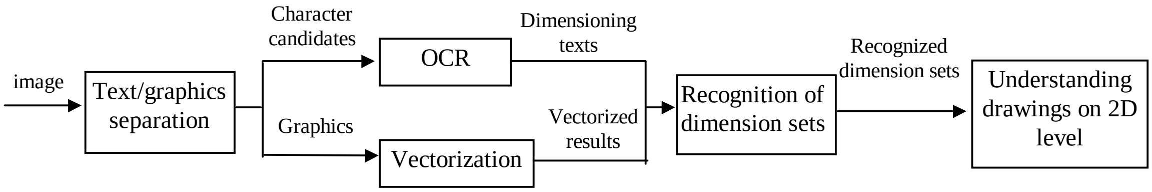


Fig. 2 Block diagram of dimension sets recognition procedure

Our method works bottom-up. First, we segment and recognize the characters in the drawings. Second, the vectorization module produces vectorized lines. We use dimension set syntaxes to recognize all the dimension lines. Then we integrate them with dimensioning text and witness lines. The dimensioning text recognition, recognition of dimension lines and integration of dimension sets are described in this section.

1. Dimensioning Text Recognition

Fig. 3 gives an example of the dimension set in architectural drawings. We can find there is no features of the dimension sets except that there is a text block near the dimension line. Our method first segments and recognizes the dimension texts. In our engineering drawing interpretation system, there is a specific module to deal with text/graphics separation and recognition[6]. The goal of this module is aimed at recognizing all the characters that are in any direction and are either human Æs writing or machine print. Reasonable as it is, it is a goal too ambitious to reach. Its average correct rate is 90%. To raise the accuracy rate, a new algorithm specially for architectural drawings which could get a higher accuracy rate, was developed. The revision to the method is described below.

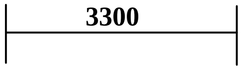


Fig. 3 Example of dimension set in architecture

A. Some criteria we use:

We can use the three criteria to extract all the character candidates.

- Connected component criteria: All the character candidates are connected components.
- Size criteria: The characters in the engineering drawings have a fixed size range. The most frequently appeared connected components should be characters, and the average size of them is charactersÆsize.
- Group criteria: The characters whose central points are in one horizontal or vertical line and whose distances to one another almost have the same value(with some tolerance) should be grouped as one string.

B. Restriction to character direction and Special pattern file to the drawings

For a machine printing architectural drawing, there are only two character directions: horizontal and vertical. Because there is no oblique string in the drawings, we could get a high accuracy rate for the extracted strings. The drawings we processed are all drawn by computer driven plotter, so the texts in them have a fixed character type. It is possible and convenient for us to create a full pattern file which covers all the character types in the drawings. During the recognition stage, a precisely matching template method was employed. By doing these, we can get a high recognition accuracy.

C. Simple touching character processing

It is difficult to handle the touching characters in the engineering drawings because they have no major difference with the graphics in the drawings. We use a projection method[7] to segment the simple touching characters. First, we relax the size criteria a little not to discard the connected components whose sizes do not exceed the character size too much. Then we define two functions as $V(x)$ and $\Phi(x)$ to find the cutting position. The position having sharp maximum value of $\Phi(x)$ is the cutting position where we can separate the touching characters. Then we use recognition engine to verify whether it is really a touching character. The verification process is a recursive one and it may test several cutting positions to get the correct character recognition results. The two functions of $V(x)$ and $\Phi(x)$ are defined as below. Fig. 4 shows one example of them.

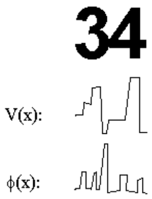


Fig. 4 Examples of functions of V(x) and Φ(x)

V(x): The function mapping horizontal position to the number of blob pixels in vertical column at that position.

Φ(x): $V(x-1) - 2 \times V(x) + V(x+1) \qquad \left(1 < x < Width\right)$

2. Recognition of Dimension Lines

In [4], Kasturi et. al describe a model-based procedure for detecting arrowheads. Das and Langrana[5] developed an image segmentation process based on morphological filters to recognize arrowheads. In both of the works, dimension lines are detected based on the recognized arrowheads. Because of the special shape of arrowheads in the architectural drawings, their methods are useless here.

The algorithm we have designed is capable of handling the dimension lines in the architectural drawings. First, our system extracts and recognizes the characters and vectorizes the other parts of the drawing. Secondly, our algorithm uses the syntaxes the geometric objects yield to in the engineering drawing and a data structure named Planar Position Index to recognize the dimension lines.

Fig. 5 shows recognized characters and vectorized results to the example drawing shown in Fig. 1. In our present work, we only recognize the two surrounding layers of dimension sets(both horizontal and vertical ones). They all belong to the type of dimension sets shown in Fig. 3. Some shape and location feature of the dimension sets are useful for us to identify the dimension lines. We can define a dimension line as a line with a text block as dimensioning text attached at its back and lines as the witness lines perpendicular to it at its ends. Based on this definition, we can start from the recognized dimensioning texts to find the dimension lines.

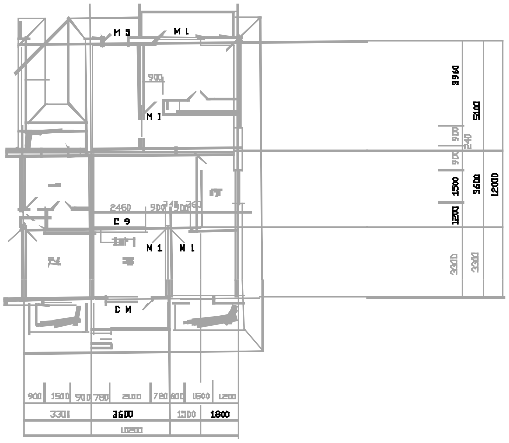


Fig. 5 Recognized characters and vectorized results

A. Grouping of dimensioning texts

The orientation of a dimension line is the same as the orientation of the dimensioning text. The orientation information of dimensioning texts can be used to search the dimension lines they attach at. We classify the dimensioning texts as horizontal and vertical, according to their orientations. Two arrays, namely vert_group and horz_group are then created. The dimensioning texts in the horz_group are stored in an ascending order of the y_coordinates of their centroid points. The dimensioning texts in the vert_group are stored in an ascending order of the x_coordinates of their centroid points.

B. Recognition of the outmost horizontal dimension lines

In Fig. 3, the outmost horizontal dimension set is special. Its dimensioning text is ô10200ö, but its dimension line is crossed by other three vertical lines. The special way to recognize it is designed as below.

To the last dimension text whose y_coordinate of the centroid point is the biggest, a rectangle area is constructed under it to search the dimension line. A horizontal line which meets two vertical lines at its ends will be found. Then we search for another horizontal line at its ends to extend the line. The work continues at both of its ends until no horizontal line can be found. We regard all the horizontal lines as one horizontal line and record the long line Æs starting and ending y_coordinates. We recognize the first dimension line which is located at the most outer layer.

C. Recognition of dimension lines at the second outmost horizontal layer

After removing the dimensioning text whose y_coordinate is the biggest from the horz_group, we can get several dimensioning texts whose y_coordinates of the centroid points almost equal to one another and bigger than other dimension texts. For these dimension texts, rectangle searching areas will be constructed under them and horizontal lines which meet two vertical lines at their ends will be found. No extension operation needs to do on this layer. We record each line Æs start and end points Æposition and then recognize the dimension lines at the second layer.

D. Error detection

After we get the outmost two horizontal layers of dimension sets, we have the opportunity to detect the errors introduced by previous steps. We calculate the sum of the dimensioning texts at the second outmost layer to verify whether it equals the outmost layerÆs dimensioning texts. If they do not equal to each other, an error alarm should be created.

We can use the similar method to find the vertical dimension lines which are located at the outermost two layers.

3. Integration of dimension sets

In other works[4,5], based on recognized dimension lines, a search rectangle was constructed to search for the dimension text which are either recognized or manually created. Then leaders(the tail combined with the arrowhead is called leaders), witness lines and associated dimension texts are integrated. That is to say, dimension sets are recognized.

Our method works in the reverse direction. According to the information of dimension texts Æ locations, we recognize the dimension lines and witness lines then integrate them as dimension sets. We can say we integrate the dimension sets at the same time as we recognize the dimension lines. So our method is more efficient in the integration period than other methods.

The property we are most interested at is the locations of the dimension sets in the drawing. We can know the real lengths for some lines in the drawings by this information. So, we just record the start pointsÆand end points Æcoordinates and the dimension set values as the recognized dimension set results. The results of recognized dimension sets in Fig. 5 are shown in Table 1 and Table 2. After these works, we have recognized the dimension sets at the surrounding two layers from the scanned drawing image, which serves the basis for our further application in the architectural drawings.

Dimension set Number	Dimension set value	Starting x_coordinate	Ending x_coordinate
0	10200	136 ± 5	2288 ± 5
1	3300	136 ± 5	512 ± 5
2	3600	512 ± 5	945 ± 5
3	1500	945 ± 5	1122 ± 5
4	1800	1122 ± 5	2288 ± 5

Table 1 Recognized horizontal dimension sets results

Dimension set Number	Dimension set value	Starting y_coordinate	Ending y_coordinate
0	12000	203 ± 5	1634 ± 5
1	5100	203 ± 5	820 ± 5
2	3600	820 ± 5	1242 ± 5

3	3300	1242 ± 5	1634 ± 5

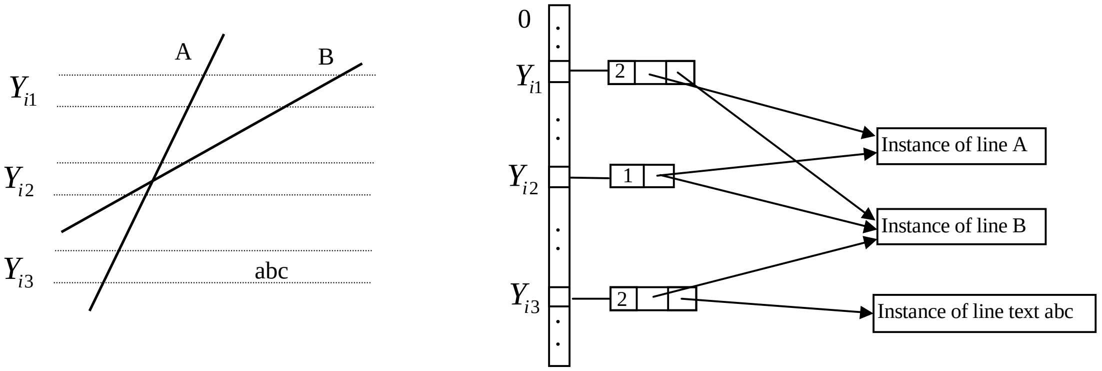
Table 2. Recognized vertical dimension sets results

III. Data Structure and its evaluation

During the recognition and integration period in our work and in [4,5], a position-to-object search was employed for many times. Sequential lists are normally used to record objects. With no geometric planar position relations among objects, searching an object in a given area require sequential search of the object lists. This is inefficient because to check for the presence of some object we want to avoid searching objects in the whole drawing, when only a much smaller area of the drawing is of interest.

1. Planar Position Index

To facilitate geometric object search in a given area, a new data structure named Planar Position Index(PPI) [8]was constructed to the vectorized objects. The entire drawing area is divided into adjacent horizontal strips of equal width. Each strip has a strip number and each strip contains a node list holding zero or more nodes. Each index node is a rectangle area in the drawing with a left boundary and right boundary. The nodes in each strip are sorted by their boundaries. The sorted index node sequence is realized as a balanced tree(B-tree). Fig. 6 is an example of the part of the drawing and its data structure.



(a). Part of Engineering drawing

(b). PPI construction for (a)

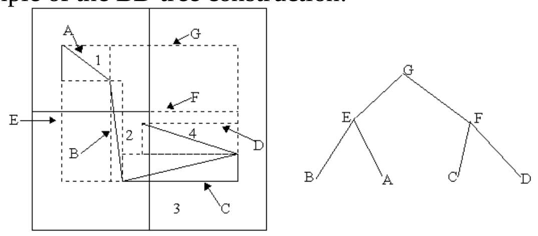
Fig. 6 Part of Engineering drawing and its PPI construction

2. BD tree

In a drawing processor named AI-MUDAMS developed by Masao Sakauchi and Yutaka Ohsawa[9], they developed a data structure named BD tree to deal with vector data. This data structure is similar with our method in many ways. A brief description of BD tree can be found below.

On the BD tree, each vector data is represented by its centroid point and its circumscribed quadrilateral. The plane in which centroid points exist is divided into two equal-size sub-plane successfully. The axis of the division is selected alternatively from vertical and horizontal axes. This decomposition process continues until just one centroid point data exists in each divided sub-plane.

Fig. 7 shows one example of the BD tree construction.



(a)

(b)

Fig. 7 part of the drawings and its BD tree construction

3. Comparison

In both of methods, searching an objet at some position is more efficient than normal sequential searching method. The time complexity of searching an object in PPI is $O(n \log n)$, where n is the average number of nodes in a strip. The time complexity of searching an object in BD tree is no less than $O(N \log N)$, where N is the number of all the vector objects in the drawing. The time complexity is larger than $O(N \log N)$ when the tree is not balanced, which depends on the contents of the drawings.

To the space complexity, PPI records one object several times(when it passes more strips than one strip), but BD tree records one object only once. So the space complexity of PPI is larger than BD tree.

As a combination of consideration of time and space complexity, the time complexity seems more important than space complexity for the drawing processor and the larger space occupied by PPI is tolerant. So PPI is more reliable and efficient than BD tree.

IV. Building Costs Estimation

After we recognize the dimension sets in the architectural drawing, we have the opportunity to do some understanding of the drawings on 2-D level. The application mentioned here comes from a real application background. Before an architectural engineer begins a new building project, he always wants to estimate the building costs of part or whole of the building. The building costs can be gotten by a definite calculation on the area of the building zone. So an automatic calculation of zones Æarea from a scanned architectural drawing is desirable.

By using our new dimension sets recognition methods, we can easily recognize the dimension sets which are useful for the estimation. So when one user selects on zone(always in rectangle shape) in the drawings, our program can tell the user the real parameters(e.g. the length and width for the rectangle zone) and the areas of the zones. For example, if the user selects the whole building he can pick up the upper-left point and lower-right point of the zone in the drawing. Based on the coordinates of the two points, the software can know the dimension set values on both horizontal and vertical directions are $10{,}200$ and $12{,}000$. So the user can be told by the software the zone area he selects is $10{,}200 \times 12{,}000 = 122{,}400{,}000$. In other words, we realize a 2-D understanding to the architectural drawings.

V. Implementation and future work

The system for building costs estimation was developed in C++ in MS-Windows environment and runs on PCs. Currently our system realizes the surrounding two layers of the dimension sets group, which can serve quite well to the area calculation to the building without consideration of the width of the wall, the existence of windows and some rooms in special shapes. Our current method does not pay much attention to the recognition of witness lines, so we have much difficulty in the automatic area calculation in the architectural drawings. They are also the future research works for us to do.

Reference

1. T. Kanungo, R.M. Haralick, and D. Dori. Understanding engineering drawings: A survey. In *International Workshop on Graphics Recognition,* pages 119-130, 1995
2. D. Dori. A Syntactic geometric approach to recognition of dimension in engineering machine drawings. *Computer Vision Graphics Image Process, 47,* pages 271-291, 1989
3. W. Min, Z. Tang, and L. Tang. Recognition of dimensions in engineering drawing based on arrowhead-match. In *Proceedings of the International Conference on Document Analysis and Recognition,* pages 373-376, 1993
4. C.P. Lai and R. Kasturi. Detection of dimensions sets in engineering drawings. In *Proceedings of the International Conference on Document Analysis and Recognition,* pages 606-613, 1993

5. A.K. Das and N.A. Langrana. Recognition of dimension sets and integration with vectorized engineering drawings. In *Proceedings of the International Conference on Document Analysis and Recognition,* pages 347-350, 1995
6. J. Gao, L. Tang, W. Liu and Z. Tang. Segmentation and Recognition of Dimension Text in Engineering Drawings. In *Proceedings of the International Conference on Document Analysis and Recognition,* pages 528-531, 1995
7. S. Kahan, T. Pavlidis and H.S. Baird. On the Recognition of Printed Characters of Any Font and Size. In *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. PAMI-9, No. 2, 1987
8. Liu W., D. Dori, T. Long, and T. Zesheng. Object recognition in engineering drawings using planar position indexing. In *International Workshop on Graphics Recognition*, pages 53-61, 1995
9. M. Sakauchi and Y. Ohsawa. The AI-MUDAMS, The Drawing Processor Based on the Multidimensional Pattern Data Structure. In *IEEE Computer Society Workshop on Computer Architecture for Pattern Analysis and Image Management Database - Capaidm Miami Beach, Florida*, Nov. 18-20, 1985