KNOWLEDGE-BASED UNDERSTANDING AND INTERPRETATION OF CONSTRUCTION ENGINEERING DRAWINGS

by

Yang Cao

A thesis submitted in partial fulfillment of the requirements for the degree of

Ph.D.

The Hong Kong Polytechnic University

2004

Date 25 March 2004

CHIEF SUPERVISOR

Professor Heng Li
Department of Building and Real Estate
The Hong Kong Polytechnic University

EXAMINATION BOARD

EXTERNAL EXAMINERS

Dr. C M Tam
Department of Building and Construction
The City University of Hong Kong

Professor Mustafa Alshawi
Institution: School of Construction and Property Management
University of Salford

INTERNAL EXAMINERS

CHAIRMAN

Professor Andrew Baldwin
Department of Building and Real Estate
The Hong Kong Polytechnic University

CHIEF SUPERVISOR

Professor Heng Li
Department of Building and Real Estate
The Hong Kong Polytechnic University

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written nor material which has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

(Signed)

Yang Cao (Name of student)

ABSTRACT: In the construction industry, many aspects from structural analysis to drawing production have already been computerized, except quantity surveying which includes the measurement of steel reinforcement used in reinforced concrete, still involves large amount of manual processes. Taking off is a very time consuming process. For example, at the tender preparation stage, it normally takes 4 to 5 man-months of an experienced Quantity Surveyor to complete the measurement work of a reasonable size, high-rise construction project. In order to solve this problem, this thesis presents a computer-aided quantity survey system, named as VHSTATION, to automatically recognize and interpret CAD structural engineering drawings, and to take off the amount of steel reinforcement indicated in the drawings. The methodologies integrated in the VHSTATION system include methods for automatic version control to guarantee the most update version to be analyzed; weighting symbols by the statistics of similar instances in candidate drawings under different recognition thresholds in order to adjust symbol recognition order; detecting walls in an architectural plan based on door symbol recognition; automatically extracting geometric features of architectural objects and converting the features into recognition rules, and utilizing Virtual Reality (VR) enabled 3D reconstruction and collision detection techniques to automatically identify and minimize design errors. The integration of these methods not only makes a useful contribution to the task of developing intelligent computer systems to automate the task of quantity surveying, but also provides interesting insight into the research domain of engineering drawings recognition.

PUBLICATIONS ARISING FROM THE THESIS

Journal papers

1. **Yang Cao**, Shuhua Wang, Heng Li, (2002) Automatic Recognition of Tables in Construction Tender Documents, *Automation in Construction*, Vol. 11 (5), pp. 573-584.
2. **Yang Cao**, Heng Li, (2003) Skew Detection and Correction in Document Images Based on Straight-Line Fitting, *Pattern Recognition Letter*, Vol. 24 (12), pp. 1871-1879.
3. **Yang Cao**, Heng Li, Shuhua Wang, (2003) A Novel Wall Detection Algorithm Based on Door Recognition, Submitted to *Pattern Recognition Letter*, Tentatively accepted.
4. Shuhua Wang, **Yang Cao**, Ruoyu Yang, Shijie Cai, (2002) A Rule-Based System for Automated Quantity Determinations of Steel Reinforcement in Concrete Structures, *Automation in Construction*, Vol. 11 (5), pp. 607-616.
5. Shuhua Wang, **Yang Cao**, Shijie Cai, (2001) Using Citing Information for Logical Structure Understanding of Document Images, *International Journal on Document Analysis and Recognition*, Vol.4, No.1, pp. 27-34.
6. Ruoyu Yang, **Yang Cao**, Heng Li, Qinping Shen, Automatic Acquisition of Rules for 3D Reconstruction of Construction Components, *Automation in Construction*, Accepted for Publication.
7. Heng Li, **Yang Cao**, (2004) Semiautomated Detection of Design Errors in 2D Drawings Using 3D Reconstruction. *Computer-aided Civil and Infrastructure Engineering*, 19, pp. 357-364.
8. Shuhua Wang, **Yang Cao**, Zuo Li, Shijie Cai, (2002) Approach to Page Segmentation and Classification, *Journal of Computer-Aided Design & Computer Graphics*, Vol. 14, No. 1:17-20. (EI index, In chinese)
9. Shuhua Wang, **Yang Cao**, Zuo Li, Shijie Cai, (2002) A Knowledge-based approach to logical structure analysis and understanding for multi-page documents, *Computer Applications and Software*,Vol. 19, No.4, pp.33-37. (EI index, In chinese)
10. Ruoyu Yang, Jia Hu, **Yang Cao**, Shijie Cai, (2002) A Highly-efficient Constraint-Network-Based Approach for Symbol Recognition of Engineering Drawings, *Journal of Computer-Aided Design & Computer Graphics*, Vol 14, No. 9, pp. 829-834. (EI index, In chinese)
11. Jia Hu, Ruoyu Yang, **Yang Cao**, Shijie Cai, (2002) 3D Reconstruction Technology for Architecture Structure Based on Graphics Understanding, *Journal of Software*. Vol. 13, No. 9, pp. 1873-1880. (EI index, In chinese)

Conference papers

1. **Yang Cao**, Heng Li, 2002, A New 3D Reconstruction Technique and Its Application in Measuring Building Quantities, Advances in Building Technology, Dec. 2002, HongKong
2. **Yang Cao**, Heng Li, Jia Hu, 2003, Symbol Recognition Errors Reduction Using Recognition Order Adjust Algorithm, International Workshop on Graphics Recognition Conference, 2003, Spain
3. **Yang Cao**, Heng Li, Shuhua Wang, (2004) A Novel Wall Detection Algorithm Based on Door Recognition, International Conference on Construction Information Technology (INCITE 2004), Langkawi, Malysia, 18-20, Feb. 2004.

ACKNOWLEDGEMENTS

The research is supported by a postgraduate studentship provided by The Hong Kong Polytechnic University. I would like to express my gratitude to Professor Heng Li, my chief supervisor, for his enthusiasm, his kind advice, numerous constructive comments and the revisions of my writings. I am deeply grateful to his guidance, support and friendship, and I am deeply impressed by his rigorous scholarship and his scholarly research works. The research cannot be finished without his huge support and contribution. I also would like to express my sincere gratitude to Mr. Conrad T. C. Wong for his comprehensive support throughout the completion of this research project. I am also deeply impressed by his strong commitment to research and development in the construction industry. Furthermore, I want to thank all the staff of VHSoft Technology for their friendly assistance. I would like to take the opportunity to thank my parents for their support, and patience, especially to my mother for her constant caring. Finally I wish to thank my wife Shuhua Wang, without her support, this dissertation would not have been possible.

TABLE OF CONTENTS

LIST OF FIGURES

Number *Page*

LIST OF TABLES

LIST OF ABBREVIATIONS

AAG	Attributed Adjacent Graph
ARG	Attributed Relational Graph
AI	Artificial Intelligence
CAD	Computer-aided Design
CDT	Change Description Table
CLS	Closed Line Series
CSG	Constructive Solid Geometry
DD	Discriminating Description
DG	Discriminating Graph
EDUI	Engineering Drawings Understanding and Interpretation
FGP	Feature Graphics Primitives
FLS	Feature Line Series
GP	Graphic Primitives
IBR	Inner Boundaries of Rooms
RAG	Region Adjacency Graph
SRP	Symbol Recognition Priority
VR	Virtual Reality

Chapter 1

INTRODUCTION

1.1 Introduction

It is often said that a picture paints a thousand words. At present, engineering drawings generated by Computer-aided Design (CAD) systems are widely used in the construction industry. Most CAD drawings used in the construction industry are 2-dimensional geometric representations of construction components. Engineers have to use their professional knowledge and expertise to interpret drawings and mentally transform the information presented in the drawings into useful information. Owing to these reasons, there are still many aspects where human efforts are needed in order to extract useful information from engineering drawings. This is particularly true in the case of quantity surveying which still involves a large amount of manual process, irrespective of using paper-based or electronic drawings. The target of the research is the conversion of engineering drawings in the construction industry to high-level representations with sufficient information for later stage analysis.

In order to develop a computer system to automate the taking off process, there is a need for a system that can automatically understand and interpret drawings in order to capture the semantic meaning of graphical entities presented in the drawings. Dym (1988) and Han (1997) addressed the importance of drawing understanding and interpretation in the construction industry and stated "a computer system which could automatically recognize individual elements by interpreting patterns of lines, arcs and strings would be an interesting and challenging task".

1.2 Objectives of the Research

The overall aim of the research are to develop a conceptual framework for implementing a computer-aided quantity survey system that can recognize and interpret CAD structural engineering drawings, and can take off the amount of steel reinforcement shown in the drawings. In order to test the conceptual framework, a computer system is implemented. Specifically, research objectives are to (see Fig. 1.1):

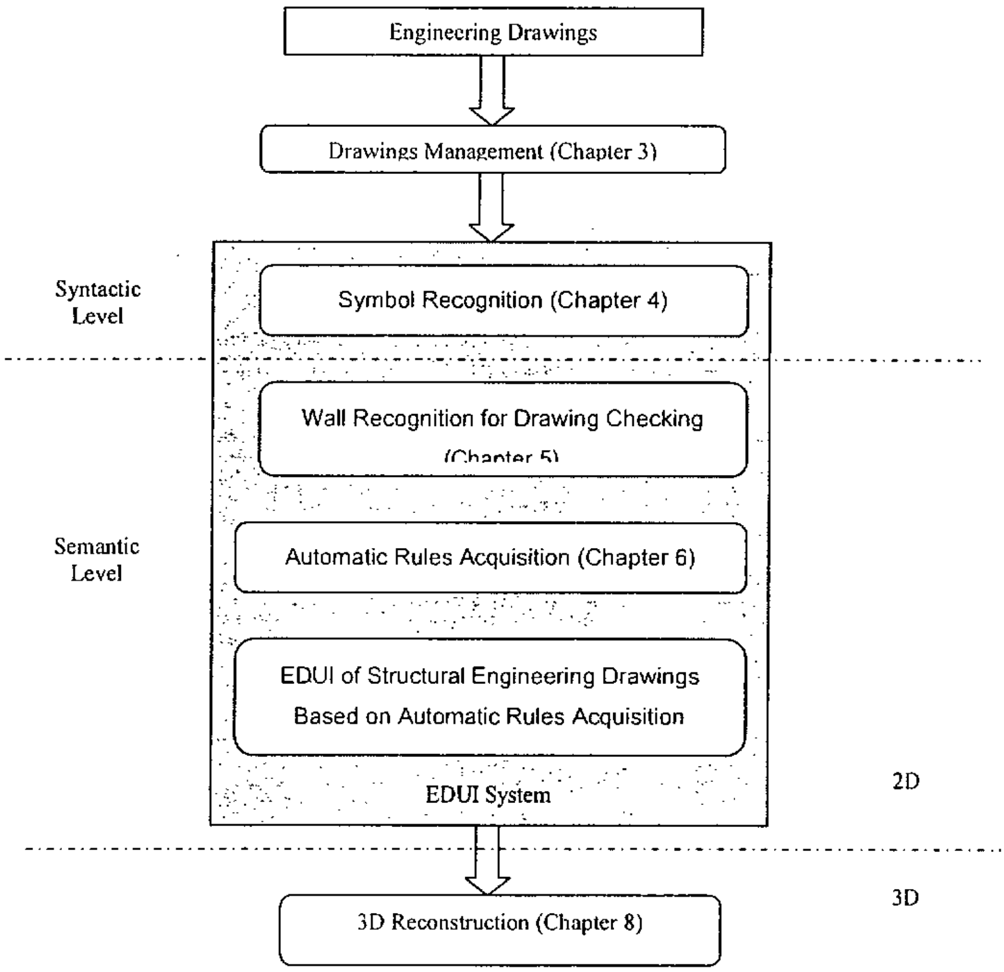


Fig. 1.1 Architecture of the Research

- research and apply Engineering Drawings Understanding and Interpretation (EDUI) techniques to the problem of automating the quantity survey work of steel reinforcement;
- build a computer system for this purpose, and use it as an experimental tool to conduct research in EDUI techniques;

- redesign and modify algorithms, or create new algorithms, to achieve improved performance. The algorithms includes:
  (1) An algorithm for automatic version control;
  (2) Symbol recognition order adjust algorithm;
  (3) Door based wall recognition algorithm;
  (4) An algorithm for automatic acquisition of rules for EDUI
  (5) An algorithm for checking design errors based on 3D reconstruction and EDUI technology
- provide experimental data to enable the computer science and informatics research communities to conduct further research in the area of EDUI toward the development of algorithms, tools and techniques for other applications.

Thousands of engineering drawings are designed, produced and used in a construction project. Different parties create drawings representing different functions of the same building. If one drawing is modified, the drawings of other related parties should also be altered. So it is important for an EDUI system to have a mechanism to monitor drawing changes and identify which versions are the most up-to-date. Chapter 3 introduces a knowledge-based algorithm for drawing layout analysis and automatic version information extraction.

The interpretation and understanding of engineering drawings can be classified into five levels: pixel level, graphics primitives level, symbol level, element level and frame level. Symbol recognition is the kernel process of engineering drawings interpretation and understanding. The element level and frame level interpretation cannot be performed without recognition of the required symbols. In the EDUI system presented, a graph-based symbol recognition method introduced by referenced paper is selected. In order to improve the accuracy and efficiency of algorithm, a method for weighting symbols by the statistic of similar instances in candidate drawings under different recognition thresholds is introduced in chapter 4.

A wall is an important element in architectural and structural drawings. It is a basic requirement of quantity surveying systems to extract wall information. The quantity of non-structural walls can only be recognized from architectural drawings, and wall information in architectural drawings describes the essential spatial information and dimensional information of a building that is important for the quantity surveying of structural elements. This is considered in chapter 5 which presents an automatic method to extract wall information from architectural drawings.

Then an element level recognition rules extraction method is presented in chapter 6. Previous work in this area can only focus on symbol level automatic rule learning for drawing recognition. The method proposed is a sample-based self-learning algorithm and the processing procedure can be described as: Designation of the sample, feature extraction, then conversion from feature to recognition rules.

The system design and implementation are introduced in chapter 7. All the algorithms and methods presented in chapter 3,4,5,6 are adopted in the system. The purpose of the system is to aid automatically recognizing and measuring steel reinforcement. It has now been fully implemented and commercially available. This is the major achievement of the research. It is shown diagrammatically in Figure 1.1.

1.3 Research Methodology

In this study, an automatic EDUI system is designed and implemented using the IPDA (Implementation, Prototyping, Design, and Analysis) research methodology (Lee 1991).

The IPEA Methodology includes three classical phases of Analysis, Design, and Implementation. In addition, it includes Prototyping as an intermediate phase situated between Design and Implementation, and dedicated to the verification and critical assessment of the analysis and design results.

Analysis is devoted to the collection and specification of requirements, expressed at a conceptual level. This phase focuses on modelling reality by means of semi-formal, expressive models and languages. Therefore, this phase uses conceptual models with an associated graphical representation that are well established in the software engineering practice.

Design is the process of translating semi-formal requirements into design documents that provide a precise, unambiguous, and computer-processable specification of the application. Due to the emphasis on object orientation and rules, Chimera model and language are used as the target of this phase (Ceri et al. 1993). The IPEA design process is divided into schema design, concerned mostly with types, classes, relationships, and operations; and knowledge design, further subdivided into deductive rule design and active rule design.

Prototyping is the process of testing, at a conceptual level, the adequacy of design results with respect to the actual user needs. A variety of formal transformation methods can be applied to improve the quality of the design, verify its formal properties, and transform design specifications into equivalent forms, which exhibit specific features required by the target implementation system.

Implementation is the process of mapping conceptual specifications into schemas, objects, and rules of existing database platforms.

Chapter 7.2 and 7.3 introduce how the system is analyzed and chapter 7.4 introduces the design considerations including: Development Strategies, Appropriate System Functions and friendly user interface. The implementation of the system is introduced in chapter 7.5 and 7.6.

1.4 Organization of Dissertation

There are nine chapters in this dissertation. These chapters are organized according to their relationships with the objectives of the research.

Literature relating to approaches of EDUI and previous systems of EDUI is reviewed in Chapter 2.

Chapter 3 introduces a knowledge-based algorithm for automatic version information extraction. The method analyses the layout of the drawing frame and extracts version information with the help of predefined key words. A comparison method between two versions of drawings is introduced. Differences between two versions are marked, which facilitate engineers to quickly and easily identify the changes.

Symbol recognition is the kernel process of drawing interpretation. Traditional methods only consider optimizing the algorithm of symbol recognition itself rather than the performance of the entire system. Actually the efficiency and accuracy of a symbol recognition system depends on how well it performs on recognizing the whole set of symbols instead of any particular symbol. Chapter 4 presents a method for weighting symbols by the statistic of similar instances in candidate drawings under different recognition thresholds. Using statistical results, symbol recognition order is adjusted to reduce probability of symbol recognition errors and to improve symbol recognition efficiency.

Chapter 5 presents an automatic method to extract wall information from architectural drawings. The method uses door as the seed segment to guide the recognition of inner room boundary. Then, separating walls between rooms can be extracted. The outside walls are recognized with the help of the separated walls. At last, all the extracted walls are connected with junctions. In the presented method only the lines that are the possible boundaries of the walls would be processed, thus the accuracy and efficiency of the presented method is improved.

Aiming at the rule-based steel reinforcement measurement system for structural drawings, Chapter 6 introduces a sample-based method that will automatically extract the geometric features of architectural objects and convert the features into recognition rules. When users designate the first sample of one type of objects, this method automatically analyzes and extracts the features useful for drawing recognition. Aided by a simple interactive operation, recognition rules for this type of objects are generated. When a new sample of one object is introduced, this method analyzes and modifies the features of this object, then generates new rules. The recognition system could be improved continuously without modifying the program by adding new types or representations of the objects.

Chapter 7 introduces the architecture and implementation of a computer system, namely VHStation, for automatically recognizing and measuring steel reinforcement of structural drawings. This system utilizes rules extracted based on the feature of structural drawings to support the interpretation of drawing elements and the relationship between the elements. The use of the rules is illustrated with various examples. An innovative method, which is called enhanced gravity field relationship method (EGFRM), is developed in order to recognize the information needed for calculating reinforcement steel. The system is fully implemented and commercially available.

Chapter 8 introduces a set of Virtual Reality (VR) enabled 3D reconstruction and collision detection techniques to automatically identify and minimize design errors. The 3D reconstruction technique can reconstruct 3D models from three 2D orthographic views. This process helps identify many dimensional inconsistencies within different orthographic views of the same object that give rise to incomplete/irregular 3D models.

The contribution of the research can be summarized as follows: an EDUI system has been developed in the Ph.D research project. In order to support the implementation of the system, five algorithms have been developed to tackle different aspects of the system. According to the experimental studies, the system provides a comprehensive software environment for the work of quantity surveyor, and it is thus expected to provide an intelligent approach to other works in construction such as design, planning and management in the further development to improve the efficiency in construction. The system has been developed by the author in C++ language, implemented and is now fully operational.

Chapter 2

LITERATURE REVIEW

2.1 Introduction

In the last three decades there has been a growing interest amongst researchers in studying the field of EDUI. Intensive research has been carried out in this field, which has resulted in a large number of technical papers and reports (Tombre 1998; Dori and Tombre 1994; Vaxivire and Tombre 1994). This field has attracted immense research interest not only because it is very challenging, but also because it provides the means for automatically processing large volumes of data in office automation, building code checking, cost estimation and other business and scientific applications.

An engineering drawing has three main functions: communication, discussion and record keeping. The graphic language of engineering drawings was developed through the use of lines, symbols and texts. An object is represented through several 2D views in an engineering drawing. Each view has two main natures: geometric nature and linguistic nature (Tomber 1995). The geometric nature represents the shape of the object with interconnected edge segments using groups of graphical primitives such as lines, arcs and circles. The linguistic nature includes groups of dimensioning lines, texts, annotations and hatchings to indicate other useful information such as dimensions, locations and materials. Understanding and interpretation of an engineering drawing is to deal with these natures (Dori et al. 1994). Specifically, the goal of drawing understanding and interpretation is to capture the required information and the attached semantic meaning of graphic entities showed in engineering drawings for further analysis.

EDUI is a subset of Graphics Recognition which is again a subset of pattern recognition. Along with the optical character recognition (OCR) and the document layout analysis, it covers a broader area of document analysis and recognition. The following are the main problems to be tackled in the task of EDUI (www.cvc.uab.es/iapr-tc10):

- Recognition of graphical primitives, shapes and symbols
- Analysis of tables and forms etc. in engineering drawing
- Analysis and interpretation of engineering drawings, logic diagrams, maps, diagrams, charts, etc.
- 3-D models reconstruction from multiple 2-D views (line drawing)
- Graphics-based information retrieval
- Performance evaluation in graphics recognition
- Systems for graphics recognition
- Automatic errors detection in engineering drawings

2.2 Approaches for EDUI

Various approaches have been applied in EDUI, which can be classified into the following categories:

- Syntactic pattern approach
- Rule-based approach
- Graph-based approach
- Statistical Approach
- Artificial neural networks approach
- Fuzzy logic approach
- Neuro-Fuzzy approach
- Genetic algorithm approach

In this chapter, three most relevant approaches are reviewed: rule-based approach, syntactic pattern approach and graph-based approach.

2.2.1 Rule Based Approach

It is common that human experts express their own expertise in terms of situation-action rules. Rules indicate that in a certain situation, some appropriate actions should be taken. A rule based system uses "rules" as the knowledge representation for knowledge coded into the system. Rules themselves have their origins in a variety of sources. Some are heuristics accumulated from past experiences. Others reflect complied knowledge and are based on a compilation of or extraction from the first principles of a particular domain. Still other rules are causal in nature; that is, they describe very specific cause-and-effect relationships. Thus rules can represent several different kinds of knowledge. Rules typically take the form of if-then statements. This is a popular and intuitive knowledge representation. Constraint knowledge, which identifies a set of conditions or limits, is easily represented using rules. Another form of knowledge, pattern matching, is also a good candidate to be implemented using rules.

Rule-based systems are adaptable to a variety of problems. In some problems, information is provided with the rules and the artificial intelligence (AI) follows them to see where they lead. An example of this is medical diagnosis in which the problem is to diagnose the underlying disease based on a set of symptoms (the working memory). A problem of this nature is solved using a forward-chaining, data-driven, system that compares data in the working memory against the conditions (IF parts) of the rules and determines which rules to fire.

In other problems, a goal is specified and the AI must find a way to achieve that specified goal. For example, if there is an epidemic of a certain disease, this AI could presume a given individual had the disease and attempt to determine if its diagnosis is correct based on available information. A backward-chaining, goal-driven, system accomplishes this. To do this, the system looks for the action in the THEN clause of the rules that matches the specified goal. In other words, it looks for the rules that can produce this goal. If a rule is found and fired, it takes each of that rule's conditions as goals and continues until either the available data satisfies all of the goals or there are no more rules that match (Clive 1991).

Although many approaches of graphics recognition have been applied to EDUI, it is very difficult for a person without domain knowledge to develop and use a hybrid system between EDUI approaches and domain knowledge. The reason is that most graphics recognition approaches require computer-programming skills in order to apply knowledge to the system. It is not a simple process. However, a rule-based system has been the most common technique in knowledge-based AI for a hybrid system with EDUI because it has a very simple rule structure to derive an answer. The rules for an application can be easily constructed in the if-then structure by a non-expert once the rule-based engine (or interpreter) is developed.

Early systems such as DENDRAL (Buchanan et el., 1969) and MYCIN (Shortliffe, 1976) demonstrated the success of rule-based approach. After these successes, many systems have been developed for technical applications of various domains. Many systems of EDUI have also adopted the rule-based approach.

Henderson (1984) introduced a method to extract swept subtractive features such as cylindrical holes, pockets and slots using logic programming rules. With this method, a 3D model in boundary representation is converted into facts in PROLOG. These facts are used by production rules that encode the necessary and sufficient conditions for a feature.

Niyogi and Srihari (1986) used three kinds of rules in a production system for document understanding. The document image is firstly segmented and descriptions about various regions are obtained. Knowledge rules pertain to intrinsic properties and spatial relationships among various regions. Control rules are used to decide which knowledge rules need to be executed. Strategy rules are used to determine whether a consistent interpretation has been obtained.

University of Twente developed a commercial process-planning system PART (Planning of Activities, Resources and Technology) (Houten et al. 1989). PART incorporated a rule –based feature recognition system. The patterns to be identified as features are specified in a feature description language. Feature recognition has two phases, feature pattern recognition and parameter extraction.

Yuhong and Ashok (1997) presented a system for recognizing a large class of engineering drawings. The class includes several domains such as flowcharts, logic and electrical circuits, and chemical plant diagrams. The automatic recognition task is divided into two stages: 1) Domain independent rules are used to segment symbols from connection lines in the drawing image that has been thinned, vectorized, preprocessed in routine ways. 2) A drawing understanding subsystem works in concert with a set of domain-specific rules to classify symbols and correct errors automatically.

A Rule-based approach has several advantages compared with other approaches of graphics recognition (Chadwick and Hannah, 1986, and Watson, 1997). First, rules can be easily understood because the control structure is relatively simple as an if-then format. Second, rules can encapsulate an important part of knowledge in which rules can model a complex problem. And, knowledge can be saved by rules. Third, the control structure of a rule seems to mimic some human problem-solving strategies, and hence rules saved in a system can be used for solving human problems. Fourth, rules are independent of each other. Even though some parts of the rules are deleted or incorrect, these mistakes will not affect other rules or functions. However, some mistakes of other programming languages describing an if-then structure may affect the entire function. Fifth, rules can be placed in any order in a program. Last, natural languages can be used for questions on the user interface, and hence it is very convenient for non-experts to use rule-based expert system for solving problems.

Though rule-based approach is simple and useful, there are still some limitations (Slade, 1991, and Watson, 1997). First, the number of rules about a feature is not unique and it is impossible to encode all the properties about all the different occurrences of a feature (Qiang et al. 1997). Second, the number of rules will increase when the feature's complication or the number of feature increases. The rule-based system needs rules for every candidate recognizing instances. Third, it is difficult for a rule-based system to perform an exact matching. Intersections of rule sets between two features may cause systems error recognition results. Fourth, although a rule assumes that there is a generally accepted body of explicit knowledge in a domain, there are many domains in a real world that have no underlying causal models and no generally accepted explicit rules.

2.2.2 Syntactic Pattern Approach

The syntactic pattern approach, which was developed over last two decades, has received much attention and applied widely to many practical pattern recognition problems, such as optical character recognition, fingerprint recognition, speech recognition, texture analysis, 3-D object recognition, symbol recognition and EDUI (Fu, 1982; Qiang et al., 1997; S. Collin et al., 1994; Ah-Soon et al., 1998; Wang 1992). The syntactic pattern approach makes explicit use of knowledge about the structure of the object. It utilizes a pattern description where each pattern is divided into sub-patterns called primitives. Each primitive has no direct relation to the structure of the pattern. A pattern is represented by knowledge about how sub-patterns must be combined to make up the entire pattern, and how sub-patterns relate to each other. In the syntactic pattern approaches, the analysis mode consists of constructing rules for combining primitives in order to obtain the structure of a given object. The approach is formulated around the concept of formal languages with each primitive represented as a terminal symbol and a grammar inferred for each pattern class. The recognition mode consists of a mechanism to check whether an object could be constructed using the rules associated with a specific object class. This process is called the syntax analysis or the parsing of a string. Grammar used in the approach include the finite-state grammar, context-free grammar, context-sensitive grammar, programmed grammar, indexed grammar, grammar of picture description language, transition network grammar, operator precedence grammar, pivor grammar, plex grammar, attributed grammar, etc (Huang 2002). The syntactic parsing analyses include finite-state automata, pushdown automata, top-down parsing, bottom-up parsing, Cocke-Younger-Kasami parsing, Earley's parsing, etc (Huang 2002).

Although the syntactic pattern approach is the formalization of pattern rules using pattern grammars, the rule-based method uses the rules themselves in place of the formalized language to prevent the limitations that grammar primitives impose. Since no grammar or primitives are required in a rule-based method, any definable feature concept, whether 2D or 3D, can be described by rules (Qiang et al. 1997).

One of the earliest syntactic pattern methods is proposed by Kyprianou (Kyprianou 1980). In his approach the B-Rep of a object is firstly converted into a face-edge graph with nodes representing faces and arcs representing edges of the object. Graph grammar is applied to extract features of objects.

Dori (1992) proposed a context free string grammar to recognize dimensions and size tolerances from engineering drawings. The grammar is extensive and capable of interpreting linear, angular, diametrical and radial dimensions, but the approach cannot handle the geometric tolerances or the textural callouts in engineering drawings.

Constraint network is another kind of syntactic pattern approach that is introduced by Ah-Soon to detect symbols in architectural drawings (Ah-Soon et al. 1998). The symbol model is constructed through a set of constraints on geometrical features. Symbol recognition is the procedure of verifying each constraint of symbol model in the constraint networks.

Although the syntactic pattern approach has shown some good recognition capabilities, it has its own limitations. The major limitation of syntactic pattern approach is the difficulty in learning or deriving structural rules from a set of training patterns. Its learning process may require significant human interaction. Though syntactic pattern approaches have been applied to recognizing 2D prismatic parts successfully, they have little success in 3D parts, for the lack of suitable languages to describe 3D objects in these approaches.

Another limitation is the ambiguity of the syntactic patterns (Wang 1992). The primitives involved in the syntactic approach usually cannot represent some geometric properties, such as the size of the primitive, relative orientation of the object, edge concavity, etc. Though syntactic pattern approach has been applied successfully in symbol recognition, these limitations will prevent it from being used widely in other fields of EDUI.

2.2.3 Graph-Based Approach

In graph-based approaches, the topology of an object is represented as a graph. Then the graph representation is searched for certain properties to identify the object instances in the engineering drawings. The graph approach gained momentum at the end of 1980 s, and it is one of the most prevalent recognition techniques at present (Lladós et al. 2000; Zhi et al. 2003), for it can directly use many developed concepts and algorithms from applied mathematics, especially in graph theory and topology. The graph approach can be informally classified into two categories: graphics recognition and structural pattern recognition.

Graphics recognition is a sub-field of document analysis and understanding. It deals with symbols, charts, diagrams, logos, etc. Some applications of graphics recognition are the matching of fingerprints, the interpretation of flow charts and diagrams, the analysis of musical scores for conversion to MIDI, the conversion of engineering designs to a CAD-compatible representation, the interpretation of architectural drawings, the analysis of maps in GIS environments, and the interpretation of electronic circuit diagrams, etc (Tombre 1998).

Structural pattern recognition, as a research area, started at the end of 1960's. Nowadays, it is still actively pursued as a research area and a number of applications have been proposed using this technique. The basis of the structural pattern recognition is the relational organization of low-level features into higher-level structures. This relational organization is represented by means of symbolic data structures like strings, trees, graphs or arrays instead of vectors of numbers that are used in the statistical pattern recognition approach. These data structures also allow a hierarchical organization of the information. There are two particular problems about structural pattern recognition: (1) Error-tolerant (sub) graph isomorphism. Graphs are clearly the most widely used data structure to represent n-dimensional prototypes. Thus, a pattern may be recognized by means of a graph matching procedure. Since the presence of noise and distortion in input images is a usual fact, an exact graph matching procedure appears to be unreliable. An error-tolerant model has to be applied in the matching process. (2) Inference of structural models. One of the fields where structural pattern recognition has an intensive action is structural texture analysis. A structural texture is characterized by a repetitive element called texel repeated according to a set of placement rules. Graph-based formal grammars are useful structures to represent such repetitive structures.

Maderlechner (1986) proposed a method of discriminating graph (DG) to give a solution for symbol recognition. The symbol of DG is a labeled subgraph. In general there are more than one DG for a certain symbol, comprising a hierarchically ordered set called the discriminating description (DD). Symbol recognition is performed by subgraph isomorphism of DG.

Kuner (1986) proposed an improved subgraph isomorphism method for consistent classification of symbols in engineering drawing. The problem of subgraph isomorphism is transformed to a quadratic classification problem that leads to an integer optimization solution. Thus, for the recognition of perfect symbols only partial coincidence between the graphs is necessary.

Joshi and Chang (1988) introduced a feature extraction method in engineering drawing based on attributed adjacency graph. The attributed adjacent graph (AAG) makes use of B-Rep information on faces and edges. An AAG is defined as a triple G=(N,A,T), where N is the set of nodes, A is the set of arcs, and T is a set of attribute values for arcs in A. In this method, the topological and geometric relationships for a depression feature are firstly translated into a local AAG that represents feature faces and has only concave links. Unique properties for a particular feature are subsequently extracted from the local AAG of a feature and represented in terms of heuristics that define the feature.

Messmer and Bunke (1996) provided a subgraph isomorphism for symbol recogntion system in engineering drawings. The system represented symbols and drawings by attributed relational graphs (ARG) and recognition is performed by searching for subgraph isomorphism.

Lladós and Martí (2001) designed a system for analysis architectural plans. The knowledge of the drawing is described in region adjacency graph (RAG). The analysis is performed by a subgraph isomorphism algorithm to search the instances of RAG model in segmented architectural plans.

Graph-based approach is one of the most prevalent recognition techniques, which uses many developed concepts and algorithms from applied mathematics, but there still exist some limitations. First, Graph-based approach is unable to represent and recognize objects made up of several disconnected segments. Second, computation complexities of all the graph-based approaches are very high. Graph construction and subgraph isomorphism are both computationally expensive. Third, most graph-based approaches consider only a limited set of patterns. Searching more patterns becomes complicated. Fourth, Graph-based approaches are typically weak at recognizing features that intersect. Fifth, Graph-based approaches are verbosity in terms of the number of graphs required to represent the features.

Table 2.1 is shown to summarize and compare the three main techniques.

Table 2.1 Three Main approaches

	Rule based approach	**Syntactic pattern approach**	**Graph based approach**
Knowledge Representation	Rule	Pattern	Graph
Recognition process	Rule matching	Syntax analysis or the parsing	Graph matching
Advantages	(1) Rules can be easily understood (2) Model complex problems (3) Mimic some human problem-solving strategies (4) Rules can be placed in any order in a program (5) Natural languages can be used	(1) Easy acquisition (2) Efficient device for use (3) High efficiency and accuracy in well-structured objects (4) Well variability and flexibility	(1) Graph structures have been studied extensively (2) Many existing algorithms can be reformulated in terms of graph (3) Reuse algorithms of mathematics which have been proved efficient
Limitations	(1) The number of rules about a feature is not unique (2) The number of rules will increase when the feature's complication or the number of feature increases (3) The number of rules will increase when the feature's complication or the number of feature increases	(1) Difficulty in learning or deriving structural rules from a set of training patterns (2) Lack of suitable languages to describe 3D objects	(1) Unable to process disconnected segments. (2) computation complexities are very high (3) Consider only a limited set of patterns (4) Weak at recognizing features that intersect

2.3 Previous Systems in EDUI

The goal of EDUI is to interpret the contents of an engineering drawing. Systems for EDUI can be categorized into: (1) CAD drawings understanding and interpretation (2D to 3D), (2) sketch interpretation (HCI), (3) indexing and searching engineering drawing, and (4) technical document analysis.

The Singapore Building and Construction Authority has developed an IFC (Industry Foundation Classes) based system **BP-Expert** for electronic engineering drawing checking (Wong et al. 2001). The system employs rule-based approach and CAD technologies in automating the checking of building plans for their compliance with building control regulations. And the IFC format provides the necessary capability for information sharing. It accepts CAD software prepared building plans and recognizes the floor layout plans, and derives the relationships amongst various building components and checks their compliance with the building regulations. IFC is also used for hand over architectural and HVAC designs to code checking sever. Though the checking process is an automatic process, the recognition of floor layout plans still requires significant human interaction and input. So the **BP-Expert** is a semi-automatic EDUI system.

A team at the University of Carnegie developed a system called **SEED** (Software Environment to Support Early Phases in Building Design) (Fenves et al. 1995). The goal is to provide support, in principle, for the preliminary design of buildings. This includes using the computer not only for analysis and evaluation, but also more actively for the generation of designs, or more accurately, for the rapid generation of design representations. A major motivation for the development of **SEED** is to bring the results of two multi-generational research efforts focusing on `generative' design systems closer to practice:

1. LOOS/ABLOOS, a generative system for the synthesis of layouts of rectangles (Flemming et al., 1988; Flemming, 1989; Coyne and Flemming, 1990; Coyne, 1991);
2. GENESIS, a rule-based system that supports the generation of assemblies of 3-dimensional solids (Heisserman et al., 1991; Heisserman and Woodbury, 1993).

SEED intends to provide systematic support for storing and retrieval of past solutions and their adaptation to similar problem situations. This motivation aligns aspects of SEED

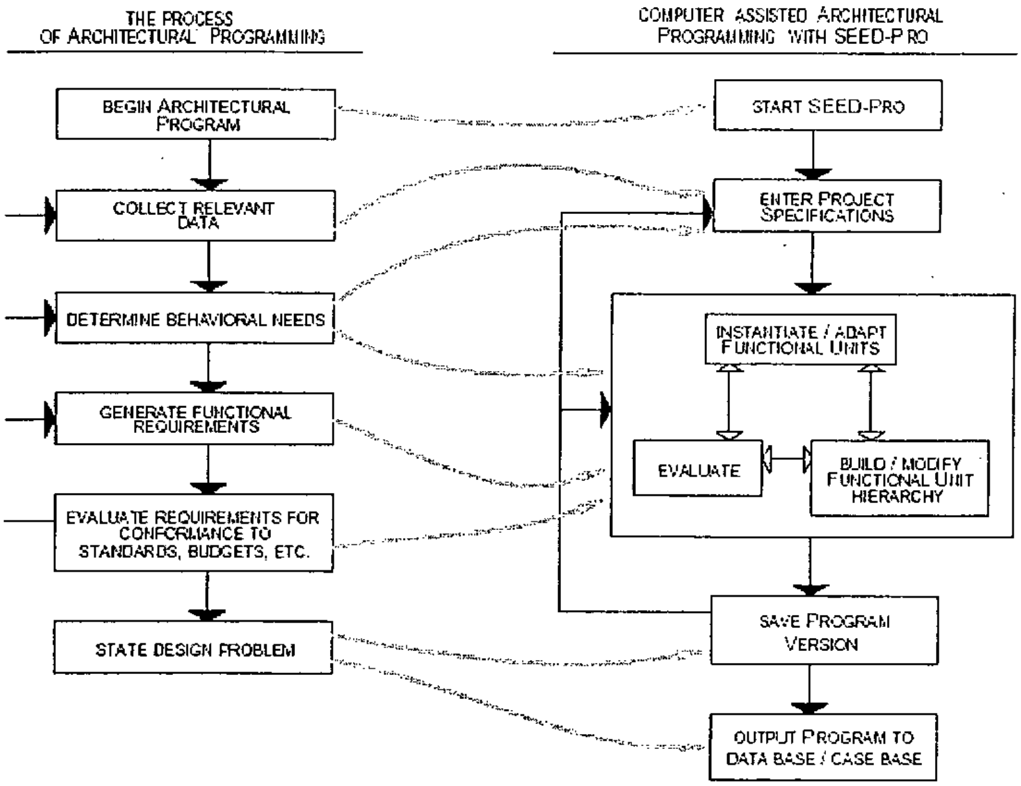


Fig. 2.1 Functions of SEED

closely with current work in Artificial Intelligence that focuses on case-based design (see, for example, Kolodner, 1991; Domeshek and Kolodner, 1992; Hua et al., 1992). **SEED** is not an EDUI system, but it is employed in some useful ideas such as using rule-based approach to analysis the relationships of design objects and drawing layout analysis (Fig. 2.1).

Lewis and Sequin of the University of California at Berkeley have developed the Building Model Generator (**BMG**) to build 3D building models from 2D architectural plans from a smoke-spread prediction model, CFAST, developed by NIST (Lewis and Sequin 1998). **BMG** utilizes room label as seed point to find interior space contour, and provides topological correction and semantic enrichment models for CFAST and WALKHRU. The approach requires the room label to be placed well in the room region. The **BMG** has reduced considerably the modeling time, it can build a model of a large building in a matter of days rather than weeks or months.

The OGAR (Querying Graphics through Analysis and Recognition) team of the lab of LORIA in France developed an engineering interpretation model **CELESSTIN** and applies it in architecture drawings to reconstruct the 3D model of a building [http://www.loria.fr/equipes/isa/index_anglais.html]. The syntactic pattern approach is used in **CELESSTIN** to recognize wall and symbol information in architectural drawings. For the inherent limitation of syntactic pattern approach, the model is not capable of analyzing the semantic meaning of the recognized objects.

From 1990 to 1995 the Hamburg AI-Laboratory has been committed to research on three successive "drawing interpretation" projects: **WIZ**, **ADIK** and **InterDok** (Gloger et. al. 1992, Pasternak et. al. 1992 (1), Pasternak et. al. 1992 (2), Pasternak 1993, Pasternak 1994, Pasternak 1995).

The **WIZ** project was part of an European cooperation project (ESPRIT 2001). The goal of WIZ was to realize a knowledge-based support for the conversion of paper-based drawings into actual CAD representations. The **WIZ** prototype focused on techniques to recognize higher-level structures in the drawing by exploiting domain knowledge. The basic processing mechanism is inspired by a blackboard metaphor, which realizes a bottom-up recognition of domain-specific objects, like dimension sets in engineering drawings (Fig. 2.2).

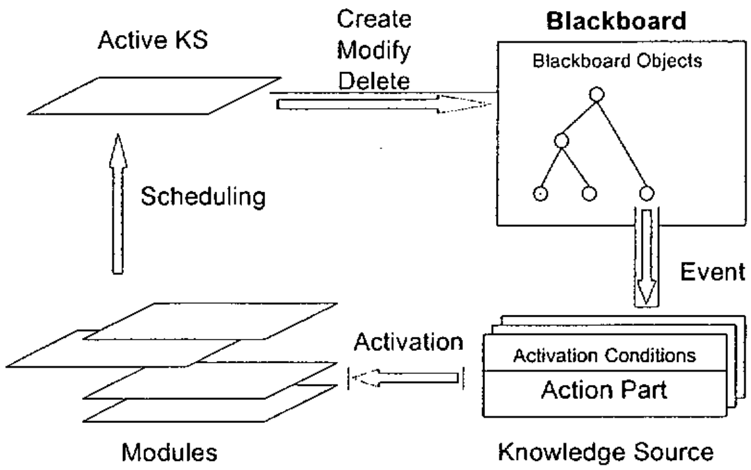


Fig.2.2 Blackboard Model

To make drawing interpretation available for other domains (like electronic circuits,

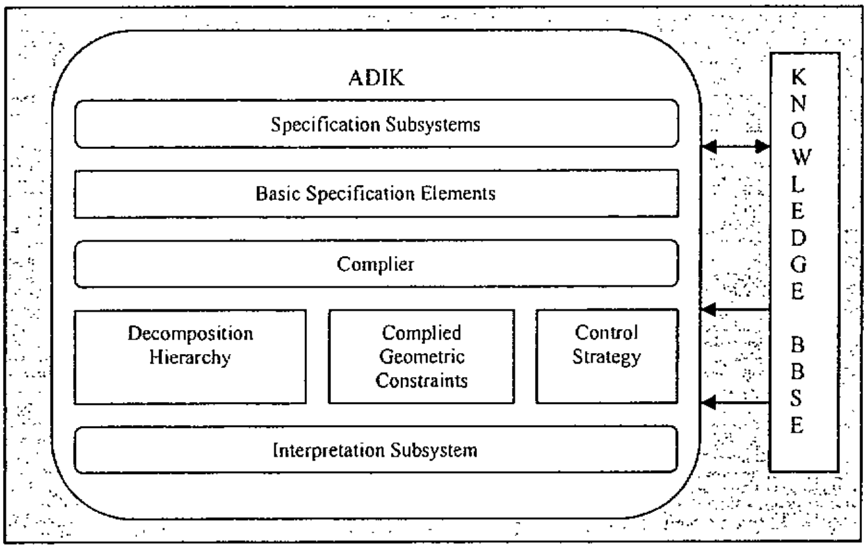


Fig.2.3 System Structure of ADIK

geographic maps, architectural plans, etc.) an "adaptable drawing interpretation kernel"

(**ADIK**) was developed as a prototype (Fig. 2.3). The key elements of this approach are a declarative specification language for geometry and a specialized inference engine that recognizes the described objects very efficiently.

InterDok is a cooperation project with *Technical University of Hamburg-Harburg, Körber AG, Daimler Benz Aerospace-Airbus, ASCAD GmbH Nord* and *SerCon GmbH.* The aim of the project is to further develop the ADIK prototype into a practically applicable tool for various "drawing interpretation" purposes.

Though the aims of **WIZ, ADIK** and **InterDok** is to provide a domain-independent drawing interpretation system, most of the approaches and models applied in systems only shows successful example in mechanical drawings and can not be used in the construction industry.

MDUS (Machine Drawing Understanding System) is a system developed by team at department of Industrial Engineering and Management of Israel Institute of Technology,

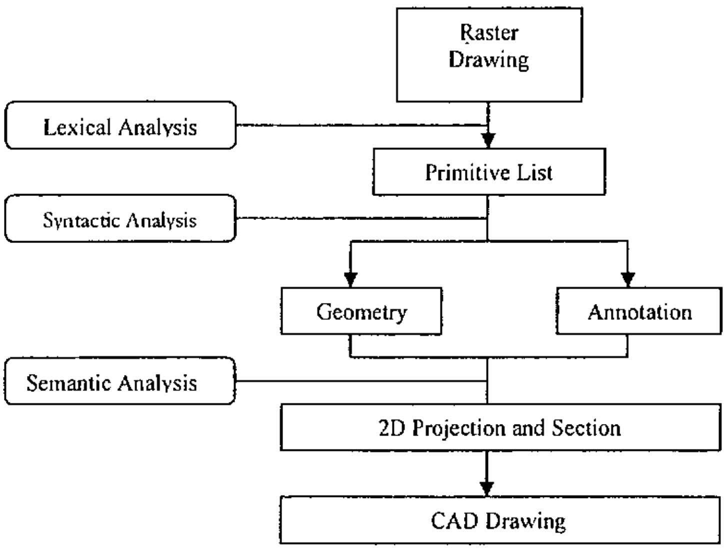


Fig.2.4 System Structure of MDUS

which is ultimately aimed at reconstructing 3-dimensional objects described in drawings (Tombre 1997, Dori. 1997). Several algorithms for sparse-pixel primitive detection were developed and implemented, including recognition of bars, arcs, arrowheads and text (Dori 1995; Dori and Liu 1999). In an international contest held in 1997, **MDUS** (Fig. 2.4) was rated as the best performing systems in the domain of mechanical engineering drawings.

Brain W. of Bell Laboratories describes a sub-system of **WISE** (Wireless System Engineering), a tool for design and optimization of indoor wireless communications systems, which is to extract information about the walls of a building from architectural plans (Kernighan et al. 1996).

Research in the area of EDUI in the construction industry is relatively new; therefore, there is little pertinent literature in this area. Only a few systems have been developed for the understanding and interpretation of constructional engineering drawings (Ah-Soon and Tombre. 1997). Most systems focus on processing architectural drawings. In a construction project, drawings can be classified into the following categories based on the functions: Site plan, Architectural plan landscaping plans, civil plans and, Structural plans. Structural plans gives the most detail information of a building compared with other plans, and is also the most complicated plan. Few algorithms or systems are proposed for structural plans or other type of engineering drawing in the construction industry. Table 2.2 summarizes the approaches.

Table 2.2 Algorithms of Drawing Understanding and Interpretation in Construction

Author and Year	*Category*	*Output*	*Approach*	*Purpose*
Koutamamis 1989, 1992,1993	CAD Drawings Understanding and Interpretation	Building Element	Template matching	Architectural design
Flemming et al. 1988,1989,1993	CAD Drawings Understanding	Rooms	Syntactic Pattern and	Architectural design

	and Interpretation		Rule based approaches	
Yessios, C et al. 1989	CAD Drawings Understanding and Interpretation	Architectural elements	Rule based approach	Architectural design
B.O. Nnaji et al. 1990	CAD Drawings Understanding and Interpretation	Architectural elements	Neutral network	Architectural design
A.J. Pollock et. Al. 1994	CAD Drawings Understanding and Interpretation	Wall, room	Graph based appraoch	Architectural design
Ryall Shieber 1995	CAD Drawings Understanding and Interpretation	Wall and Rooms	Image process and recognition	Architectural design
Brian W. 1996	CAD Drawings Understanding and Interpretation	Wall	Sweep Algorithm	Wireless communications predictions
J. Lladós et al. 1996, 1998, 1999.	Sketch Interpretation	Wall and Room, Symbols	Graph based approach	Architectural design
Y. Aoki et al. 1996	Sketch Interpretation	Wall	Sweep Algorithm	Architectural design
M Huang, et al. 1997	CAD Drawings Understanding and Interpretation	Building structure Element	Image process and recognition	Cost estimation
Christian Ah-Soon and Karl Tombre 1997, 1998	Sketch Interpretation	Wall and Rooms	Sweep Algorithm	3D reconstruction
Y. Cao 1999	CAD Drawings Understanding and Interpretation	Column, Beam Wall, Slab	Rule-Based Approach	Cost Estimation
Song Z-L et al. 2002	CAD Drawings Understanding and Interpretation	Room	Graph Approach	Room Searching
S. Wang et al.2002	CAD Drawings Understanding and Interpretation	Structural Element	Rule-Based Approach	Quantity Survey
G. S Anchez et al.2003	Sketch Interpretation	Wall	Sweep Algorithm	Building Planning Model
Yong-Bin Kwon 2003	CAD Drawings Understanding and Interpretation	Wall	Knowledge-Direct Sweep Algorithm	N/A
Zhi G.S. et al. 2003	CAD Drawings Understanding and Interpretation	Wall, Room	Graph Based Approach	Building Evacuation Model

2.4 Conclusions

This chapter presents a survey of recent research that is related to EDUI. Several systems have already been developed, for example, CELESSTIN in France and BP-Expert in

Singapore, yet there are still limitations found in their applications. These limitations can be categorized as follows. Firstly, most of the previous engineering drawing analysis systems are still at the lexical phase, and do not have the ability to semantically interpret engineering drawings. Secondly, drawing analysis systems that were developed in earlier stage all targeted a single type of drawing. However, emphasis of recent work has found to be a need for systems that can be applied to different types of drawings. Thirdly, methods described in the previous systems could only work well in strict conditions (line thickness, grid location, etc.). Fourthly, the previous systems do not have the ability of self-learning or self-adapting. Hence, a large number of rules will be required to cover every possible condition. But as the number of rules arises, the reasoning speed of the system tends to become slow. Finally, they lack performance measures and experimental protocols. These limitations motivated the writer to develop a more automatic, self contained, less complex and more robust system to interpret constructional engineering drawings.

Chapter 3

INTELLIGENT DRAWING MANAGEMENT USING ENGINEERING DRAWINGS UNDERSTANDING AND INTERPRETATION TECHNIQUES

3.1 Introduction

Every year thousands of construction drawings are produced for each construction project using computer-aided design systems (CAD). Different parties create drawings representing different functions of the same building. Since there are no convenient tools available to monitor drawing changes and identify which versions are the most up-to-date, it would be hard for contractors to obtain the most updated version of drawings, especially when one drawing is linked to several other drawings. Therefore, in order to maintain the history of different versions of drawings and version control has become a key issue in the development of an EDUI system.

The task for controlling the development of different versions of a drawing can be complex. It will become more complicated when the number of authors increases (Dix and Miles, 1992). Various attempts have been made to categorize and index drawings. While some of the methods can be very helpful for the version control, most of the methods require manually inputting necessary version data.

Standardizing key information in engineering drawing is helpful for version control, and is widely applied. For example, vendors of the Department of Defense of the U.S. government are required to provide technical publications complied with particular military standards, such as MIL-M-38784 and MIL-M-81927. CERN (CERN EST/98-

01 (ISS) CDD Manual) also adopts a method to use a unique fixed length alphanumeric number identifying the key information in engineering drawings. Drawing sequential numbers are standardized by some codes, and based on these codes main data items associated with each drawing can be extracted automatically. In addition, version information can be extracted automatically. The Hong Kong Architectural Service Department and Housing Department also adopt the similar drawing management methods.

Though standardization is a very useful method, mistakes caused by incaution of designers and different parties involved in a same project make it difficult to implement. For example in the Japanese construction industry, architects and engineers only produce preliminary drawings and general contractors and their subcontractors produce all the detail drawings (Hirasawa et al. 1999). It is therefore difficult to require the architectural companies, general contractors and subcontractors to follow the same drawing standards. In this regard, flexible methods are needed. Hughes Aircraft Company introduces an automated logistical relational database support system for engineering drawings management (Virgil et al., US patent US-5, 493,679). The system extracts version information in a drawing frame using a model-based method. Firstly, the method defines patterns that describe the extraction area of drawing frames for different kinds of drawing. When a new drawing is introduced to the system, it selects a predefined pattern most similar to the drawing frame used in the new drawing and extracts the information based on the selected pattern. But the method does not analyze the layout of version information in the drawing frame. The region and layout of the drawing frame is defined through user interaction. In addition, for engineering drawings with different layout plan of drawing frames, predefined patterns should be prepared and implanted into the system before version extraction.

In this chapter a knowledge-directed version information extraction method is presented. This chapter firstly presents a method that analyzes the layout of the drawing frame and extracts version information with the help of predefined key words. Next, a description of changes shown in the table is recognized and information relating to the description of the changes is extracted. Then a comparison method between two versions of a drawing is introduced. Dissimilar graphic primitives between two versions are marked with different colors, so that engineers can locate and identify changes quickly.

3.2 Automatic Recognition of Version Number

3.2.1 Version Number Extraction in Drawing Frame

A standard engineering drawing can be regarded as consisting two parts: drawing content and drawing frame. While the drawing content gives detailed descriptions of engineering objects, the drawing frame supplies generic information concerning properties of the drawing. Version information is part of the important information in the drawing frame.

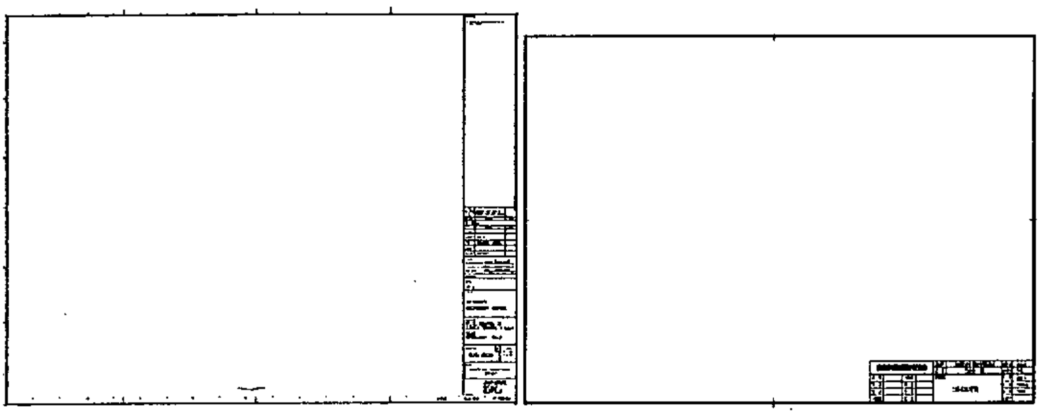
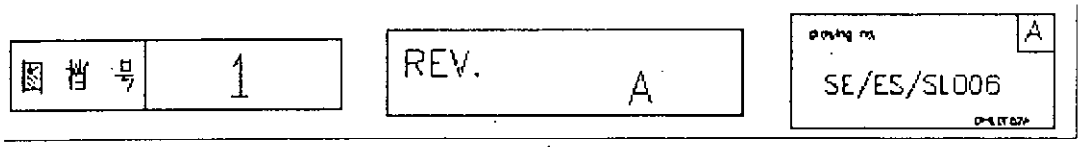


Fig. 3.1 Drawing Frame

Even though drawing content could be complicated, the format of a drawing frame is similar across all kinds of drawings (see Fig.3.1). Based on the fact, an automatic detection algorithm is designed to recognize the version information illustrated in the drawing frame. There are three typical representations for the version number information (see Fig. 3.2). In type *a* and *b*, version number is described with a key word,



a b c

Fig.3.2 Three representations of version number

and in type *c* version number is attached with other drawing information such as the drawing number. A detail description of the recognition method of version number is described in the following steps:

Step 1: Collect all the graphical elements from a CAD drawing.

Step 2:Find all the enclosed rectangular regions in the drawing. Then check all the rectangular regions and find the rectangular regions fulfill the following three constraints: (1) The rectangular is located at the right or bottom of the drawing, (2) The rectangular comprises of a group of rectangular regions, (3) Some strings in the predefined key words dictionary could be found in the enclosed rectangular. The rectangular regions fulfilled with the three constraints are the regions of drawing frame.

Step 3: Get the drawing frame and store all the inside closed rectangular regions in a Set A.

Step 4: Go through the inside area of every rectangular region in Set A to find a region that fulfills one of the following criteria (1) The region contains one string, and the string is also in the predefined key words dictionary. (2) The region contains two strings, one is in the predefined key words dictionary and the other is a character. If the region fulfills the first criterion then go to step 5. If the region fulfills the second criterion, then the character is the version number. If could not find such a region, got step 6.

Step 5: Check all the regions that are adjacent to the region found in step 4, and find the region that contains only a character. The found character is the version number.

Step 6: Find all the regions that contain a character and check whether the regions are attached to another drawing information region. If the regions can be found, the characters inside are the candidates of version no. Otherwise, the drawing number cannot be recognized until the recognition of Changing Description Table. Compare the version information found in the Changing Description Table to decide which character is the correct version number.

3.2.2 Changing Description Table Analysis

Change description table (CDT) plays an important role in version control. It is the kind of table used to describe some important information related to the changes of new version. The columns of the table could contain Data, Initial and Description of the changes (see Fig. 3.3). Many research efforts have been reported on developing methods to recognize tables (Cao 2002). Limitations of these systems include that they could only automatically or semi-automatically recognize tables using information

B	4/97	COLUMNS C26, SEC. 1-1, 2-2 REVISED TO SUIT ARCH. DRAWING.	
A	1/97	COLUMNS C44 AND C45 REVISED TO SUIT ARCH. DRAWING	SIGNED
no.	date	description	initial

Fig. 3.3 Change Description Table

stored in predefined table databases (Cao 2002, S. Chandran et. Al. 1993, Itonori 1993, Shamilian 1997). An important difference between CDT with other kinds of tables is that the column head of CDT is at the bottom of the table. It is because that the number of rows in CDT could not be determined and a new version will result in a new row.

The latest version number is stored in the top of the row. Different companies and originations have different settings of columns, so it is difficult to set a uniform model of CDT. A possible method is to set corresponding predefined patterns for every new type of CDT. But it still requires a great deal of manual input.

In order to minimize manual input, two distinctive features are used to recognize CDT automatically. The first feature is that every CDT has at least two columns: Version Number Column and Description Column. The second is that the Version Number Column contains a group of rectangular regions and each region has a character inside it and the character in the top region is the same to the version number extracted from the drawing frame. Characters inside the regions are listed in a descending sequence. Key words used for the column headings of the aforementioned two columns are limited, and could be stored in a predefined key words dictionary. The CDT detection algorithm is based on these two features and is described in details as follows:

- Find all the closed rectangular regions in the drawing frame and select the rectangular regions that only contain one character.
- Group the above regions with the width, height and position. Those regions, which have same width, height and are connected, are grouped into a set.
- Go through all the sets that are grouped in step 2 and sort the regions in the group based on the position of the character inside. For a region A, if the character inside is higher than the character of the region B, then region A is put before region B in the set. Go through all the sets and find a group that characters inside are listed in descending sequence, and record the top and bottom position the group.
- Get the groups and check whether the group fulfills the following conditions: (1) A column head region is adjacent to the bottom of the group. The column region has same height and width with the regions inside the group and contains a string that could be found in the predefined key words dictionary (2). The top most character is

same to the version number extracted before. If the group can fulfill the two conditions, a CDT in the drawing frame is recognized. The bottom and top positions of group are also the top and bottom positions of CDT and the number of regions in the group is the number of rows.

- The Description Column can also be detected based on the position of CDT and Version Number Column.

3.2.3 Drawing Comparison

To analyze the content of CDT is a task of natural language processing which is not the focus of this chapter. However, engineers can use the content in CDT to detect the differences in two different versions of drawings, as the most important concern to engineers is to identify the changes in the new version of drawing quickly. So an automatic CAD drawings comparison method is presented in this chapter to assist the engineers to detect the changing graphical primitives between different versions.

Before comparing two versions of drawings, a reference point should be detected. And all the graphical primitives are compared based on the detected reference point. Two kinds of points could be used as reference points: Intersection points of grid system and points of a drawing frame. The method used to detect the region of drawing frame is already described in this chapter. Methods for obtaining intersection points of grid system can be referred to the paper (Wang and Cao 2002). After finding the reference point, the comparing algorithm proceeds according to the following steps:

- Collect all the graphical primitive data from two CAD drawing files A, B and store them into graphic primitives sets based on drawing structure. For example Fig. 3.4 lists some typical data structure of some common graphic primitive in DWG or

DXF file that define and generate by AutoCAD (www.autodesk.com, www.opendwg.org).

Line:	Text:	Dimension:	Arc:
Layer,	Layer,	Layer,	Layer,
LineType,	Height,	Rotation,	BeginAngle ,
X0, Y0, Z0,	Font,	BlockName,	EndAngle,
X1, Y1, Z1,	Rotation,	X0,Y0,Z0,	Radius,
	X0, Y0, Z0,	X1,Y1,Z1,	X0,Y0,Z0,
	X1,Y1,Z1,	X2,Y2,Z2,	X1,Y1,Z1,
		X3,Y3,Z3,	X2,Y2,Z2
		X4,Y4,Z4	

Fig. 3.4 Typical Graphic Element Data Structures

- If A and B contain XREF data, convert the XREF data into graphic primitives sets. The term XREF is used to refer the type of data that appears in the drawing files of some CAD software that allows another drawing to be inserted at a specified point in the drawing.
- Go through every graphic primitive set and check the graphic primitive between A, B and mark the following situations: (1) A graphic element a is shown in A but not in B, (2) A graphic primitive b is shown in B but not in A (3) Graphic primitive a in A, b in B are shown in same position but do not have same properties. Properties can be classified into two types: Geometrical property and general property such as Layer, Color, font, LineType and so on.

The automatic CAD drawings comparison method is implemented in the system presented in Chapter 7. In the system two types of comparison functions are implemented based on the different reference point: comparison based on World Coordinate System (WCS), which simply compares the two versions based on their original reference point provided in the drawings, and Comparison based on User

Coordinate System (UCS), which compares the two versions based on the reference points selected by the users. After comparison, the differences are highlighted and users can locate the changes easily. A sample of comparison result is shown in Fig. 3.5.

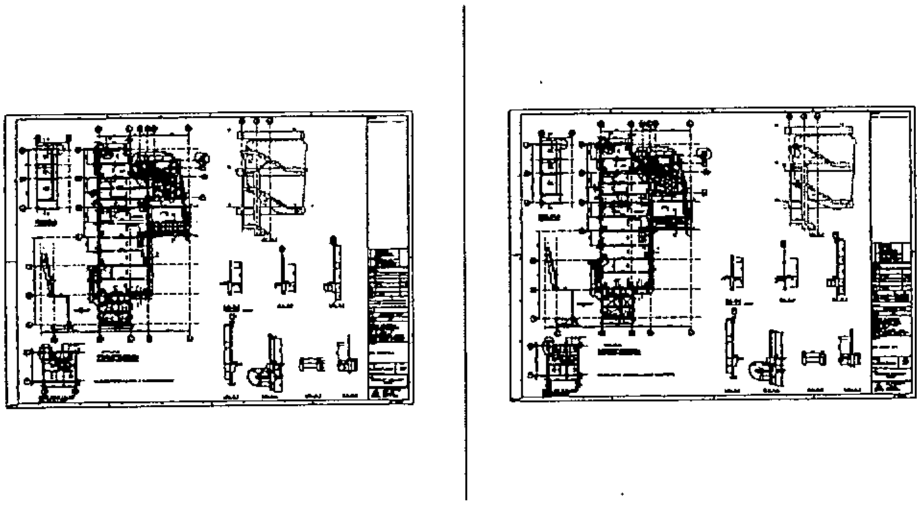


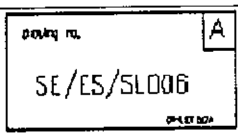
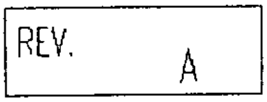
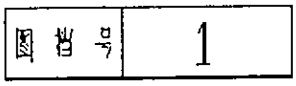
Fig. 3.5 Comparison Result of Different Versions

3.3 Experiments

This section reports two experiments on automatic version number extraction and CDT recognition conducted in the research. In the first experiment, 469 architectural drawings of three types are selected to test the algorithm of automatic version number extraction. The experimental drawings are from three sources: Hong Kong Architectural Service Department, one of Hong Kong main contractor and one of China Architectural Company. The version number information from different sources is shown in different format and the key words for identifying the version number information are also different. Except for one architectural drawing, all the other drawings had a sound result in the experiment. The only drawing that could not be processed correctly had the graphical errors in the drawing.

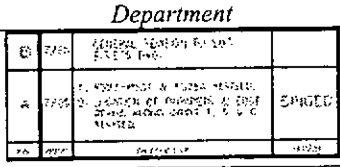
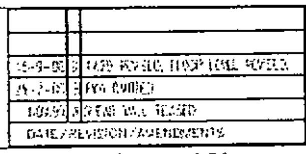
Table 3.1 Results of Version Number Extraction Experiment

Name	*Hong Kong Architectural Service Department*	*One of Hong Kong Main contractor*	*One of China Architectural company*

Version No. Type			
Type	Structure Details	Architectural Plan	Building Service Plan
Drawing Number	265	176	28
Extraction Correctness	100%	99.4%	100%

In the second experiment, 441 drawings of two types were selected to for automatic CDT recognition. Though the column fields of two types of drawing are different, the presented method can detect the CDT correctly based on the feature of version number list.

Table 3.2 Results of CDT Extraction Experiment

Name	*Hong Kong Architectural Service Department*	*One Hong Kong Main contractor*
Version No. Type		
Type	Structure Details	Architectural Plan
Drawing Number	265	176
Extraction Correctness	100%	100%

The methods proposed in this chapter are successfully implemented in the EDUI system presented in Chapter 7. Furthermore, the methods can also be widely employed in drawing management systems in construction as well as many other manufacturing sectors.

3.4 Conclusions

In this chapter, methods for automatic version control are presented. The methods not only focus on extracting the information about version but also on detecting differences between different versions of drawings. Therefore, these methods can avoid massive manual input required in other systems, and also detect inconsistencies and mistakes through comparisons of drawings. Future research will be focus on how to manage different versions of engineering drawings at the lexical and semantic level, and the user can quickly identify which architectural or construction elements have been altered, so that all the related subsequent modifications can be made automatically. Another research task will be on the reduction of storage cost on keeping several versions of drawings simultaneously. Current drawing management systems require storing all the revisions together. This makes it prohibitively expensive to run these systems in terms of storage costs, especially when no data compression techniques are employed. Finally, indiscriminately storing every change produces too many revisions, and engineers have difficulties to distinguish them. Therefore, how to use drawing interpreting technologies and comparison methods to save the changes of the new version alone will be an interesting research topic in the future.

Chapter 4

SYMBOL RECOGNITION ERRORS REDUCTION USING RECOGNITION ORDER ADJUSTING ALGORITHM

4.1 Introduction

The EDUI process can be divided into the following three phases: graphics primitive recognition; symbol recognition; and element recognition. Symbol recognition is the intermediate process of drawing understanding and interpretation. It converts discrete graphical primitives into symbols that have special meanings. Symbol recognition is the kernel process of EDUI. Most of EDUI systems have to resolve symbol recognition problems.

Many approaches have been proposed for symbol recognition. A few examples include model matching, decision tree, neural network, graph, matching, syntactic/structural matching and statistical method (Lladós et al. 1999; Ah-Soon et al. 1998; Okazaki, et al. 1988). If processed engineering drawings are not complicated, all these methods can achieve high efficiency and accuracy. However, interpreting hundreds even thousands of symbols in complicated engineering drawings is not an easy task. When the correct recognition rates of the aforementioned approaches get to a certain level, it is difficult to improve further. Currently, research efforts have been concentrated on optimizing the algorithms themselves. It is generally understood that recognition errors are caused by noises and distortions (Chhabra 1997). However, there are other types of recognition errors that are often overlooked. These kinds of errors are named as related errors, which are caused by the interference of adjacent objects in the drawing. For example, if a symbol is the sub-graph of another, it will be probably recognized wrongly. This type of error impacts not only on the symbol itself but also the recognition of other objects. Based on statistics, about 15~30% of recognition errors are related errors in symbol recognition (Cao 2003).

This chapter introduces an algorithm to adjust symbol recognition in order to reduce related errors. The algorithm counts ambiguous instances of symbols in a drawing based on calculating the similarity measured by two different thresholds that will be elaborated below.

4.2 Definition of Symbol Recognition Priority

It is generally understood that a symbol or graphical primitive that has more similar instances in the drawing will result in higher probability of error-recognition. The probability of error-recognition and recognition of a symbol depends on whether the recognition algorithm uses strict or loose thresholds. The probability of related errors occurrence between strict and loose thresholds would be higher than other fields. So in this chapter an algorithm is proposed to evaluate probability of the related errors of each symbol and recognition order is performed according to the evaluation.

The symbol model is represented in a graph structure, where graphical primitive is the node and relationships between graphical primitives are edges. The probability errors related of a symbol are the sum of the related errors probability of its nodes and edges. A similarity matrix proposed by Yeung (Yeung and Wang 2002) is used to measure the similar instances in engineering drawings of nodes and edges in the symbol model. Suppose that each node/edge is identified by a collection of features $F_j\{j = 1,\cdots,m\}$. Then for $i = 1,2\cdots,N, e_i = (x_{i1}, x_{i2},\cdots,x_{im})$, where x_{im} corresponds to the value of feature $F_j(1 \le j \le m)$. Function $Dist(p_1, p_2)$ is defined to represent the geometric similarity between two nodes or edges.

$$Dist(p,q) = \sqrt{\sum_{j=1}^{m}(x_{pj} - x_{qj})^2}$$

If $Dist(p_1, p_2) < \varepsilon$, then p_1 and p_2 can be treated as similar. The number of similar instances depends on the value of ε. In order to identify the probability of related errors of a symbol model, two thresholds value ε_1 and ε_2, $\varepsilon_2 > \varepsilon_1$ are chosen, where ε_1 is a strict threshold that tries to ensure that least errors elements will be counted into the similar set, and ε_2 is a loose threshold that tries to ensure that most similar elements will be counted into the similar set. The value of ε_1 and ε_2 could be different in different engineering understanding and interpretation systems. In this chapter a simple approach is chosen to establish the value of ε_1 and ε_2. For each recognition condition, the selected threshold is ε, then $\varepsilon_1 = 0.9 \times \varepsilon$ and $\varepsilon_2 = 1.1 \times \varepsilon$.

The definition of Density of Similar Nodes in Symbol (DSN) can be defined as:

$$DSN(n_i, \varepsilon) = \frac{S(n_i, \varepsilon)}{O(n_i)}$$

$S(n_i, \varepsilon)$ is the amount of graphic primitives similar to n_i in a drawing, $O(n_i)$ is the total amount of graphic primitives of the group which n_i belongs to in a drawing. For a symbol s with N nodes, the probability of related errors occurrence of nodes can be defined as:

$$CN(s) = \sum_{i=1}^{N} (1 - (DSN(n_i, \varepsilon_2) - DSN(n_i, \varepsilon_1)))$$

For an edge $e(n_1, n_2)$ in the symbol model, and nodes set $\{R_1 \mid n_1^i \in R_1, Dist(n_1^i, n_1) < \varepsilon\}$ and $\{R_2 \mid n_2^i \in R_2, Dist(n_2^i, n_2) < \varepsilon\}$, two edges sets could be got:

$$\{ER_1 \mid (n_1^i, n_1^j) \in ER_1, n_1^i \in R_1, Dist((n_1^i, n_1^j), (n_1, n_2) < \varepsilon\}$$

$$\{ER_2 \mid (n_2^i, n_2^j) \in ER_2, n_2^i \in R_2, Dist((n_2^i, n_2^j), (n_1, n_2) < \varepsilon\}$$

The definition of Density of Similar Edges (DSE) in Symbol is defined as:

$$DSE((n_1, n_2)) = \frac{S((n_1, n_2), \varepsilon)}{O((n_1, n_2))}.$$

$S((n_1, n_2), \varepsilon)$ returns the similar amounts of edges in ER_1 and ER_2, $O(n_1, n_2)$ returns the total amounts of the type which (n_1, n_2) belongs to in drawing.

For a symbol s with M edges, the probability of related errors occurrence of edges can be defined as:

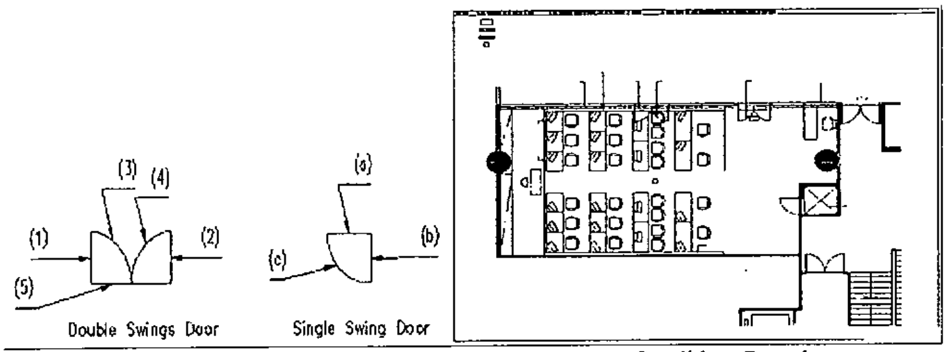
$$CE(s) = \sum_{i=1}^{M} (1 - (DSE((n_i, n_j), \varepsilon_2) - DSE((n_i, n_j), \varepsilon_1))).$$

The symbol recognition priority (SRP) can be defined as:

$$SRP(s) = CN(s) + CE(s).$$

For a set of symbols, the recognition order can be performed based on the recognition priority of each symbol.

An example is given how to calculate the recognition priority of two symbol models: Double Swings Door and Single Swing Door (see Fig. 4.1).



Symbols
a

Candidate Drawing
b

Fig. 4.1 Symbol Model and Candidate Drawing

Table 4.1. Double Swings Door Recognition Priority

Double Swings Door	(1)	(2)	(3)	(4)	(5)	(1,5)	(5,2)	(2,4)	(3,4)	(1,3)	(1,2)
$S(n_i,\varepsilon_1)$	31	31	4	3	10	137	137	8	3	8	58
$S(n_i,\varepsilon_2)$	54	54	7	7	15	286	286	8	3	8	154
$DSN(n_i,\varepsilon_2) - DSN(n_i,\varepsilon_2)$	0.973	0.973	0.625	0.5	0.994	0.998	0.998	1	1	1	0.999
SPR	10.06										

Table 4.2 Single Swing Door Recognition Priority

Single Swing Door	(a)	(b)	(c)	(a,b)	(b,c)	(c,a)
$S(n_i,\varepsilon_1)$	31	31	3	470	8	8
$S(n_i,\varepsilon_2)$	54	54	7	1147	8	8
$DSN(n_i,\varepsilon_2) - DSN(n_i,\varepsilon_2)$	0.973	0.973	0.5	0.992	1	1
SPR	5.438					

Graphic primitives in engineering drawings can be classified into the following classes: Line, Arc, and String. For statistical purposes each kind of graphical primitive is further classified into several groups. For example, lines can be classified into three groups: Isolated Line, Open Ended Line and Enclosed Line (Maderlechner 1988), and according to the classification of lines, arcs can also be classified into four groups:

Isolated Arc, Open Ended Arc, Enclosed Arc and Enclosed Loop Arc. Strings can be classified into three groups: Digital word, normal word and sentence. Different groups of graphical primitives have different recognition complexity level. Based on the grouping, the number of {Isolated Line, Open Ended Line, Enclosed Line}, {Isolated Arc, Open Ended Arc, Enclosed Arc, Enclosed Loop Arc} and {Digital word, normal word, sentence} are {0,212,884}, {0, 8, 2, 3} and {0, 9, 18}, the numbers of {Line, Line}, {Line, Arc}, {Line, String}, {Arc, String}, {Arc, Arc}, {String, String} are 87908, 6184, 8568, 67, 12 and 52. Double swings door has three lines, two arcs, three {Line, Line} edges, six {Line, Arc} edges and one {Arc, Arc} edges. Single Swing Door has two lines, one arc, one {Line, Line} edges, two {Line, Arc} edges. The SPR value of Double Swings Door and Single Swing Door can be found in Table 4.1 and Table 4.2. And Double Swing Doors should be recognized before Single Swing Door.

4.3 Proof of Accuracy and Efficiency Improvement

The relationship $\{s_i, s_j\}$ between two symbols model s_i and s_j can be classified into the following three types: $\{R_1 \mid s_i \subset s_j\}$ (see Fig. 4.2(a)), $\{R_2 \mid s_i \cap s_j \neq \phi, s_i \not\subset s_j, s_j \not\subset s_i\}$ (see Fig. 4.2(b)) and $\{R_3 \mid s_i \cap s_j = \phi\}$ (see Fig. 4.2(c)).

For two symbols s_1 and s_2 their individual correct recognition probabilities are $p(s_1)$

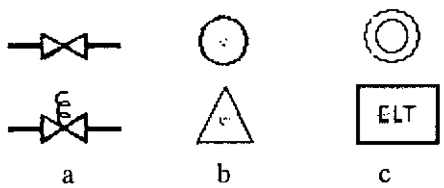


Fig.4.2 Symbol Relationships

and $p(s_2)$. If recognizing them together, the correct recognition probabilities will be $p(s_1s_2)$ or $p(s_2s_1)$ based on different recognition order.

If $\{s_1,s_2\} \subset R_1$ and $SPR(s_1) > SPR(s_2)$, based on the recognition priority, s_1 will be recognized before s_2. Errors that recognize part of s_1 as s_2 will be avoided, so $p(s_1s_2)$ will be higher than $p(s_2s_1)$.

If $\{s_1,s_2\} \subset R_2$ and $SPR(s_1) > SPR(s_2)$, the following symbols can be defined as $\{s_{12} \mid s_{12} = s_1 \cap s_2\}$, $\{s_{11} \mid s_{11} = s_1 - s_{12}\}$ and $\{s_{22} \mid s_{22} = s_2 - s_{12}\}$. Errors will occur under the following conditions: (1) For a symbol instance s_1^i in a drawing, and a similar structure s_{22}^j, if s_1^i and s_{22}^j match the constraints of s_2 and the symbol recognition system recognizes s_2 before s_1. (2) For a symbol s_2^i instance in a drawing, and a similar structure s_{11}^j, if s_2^i and s_{11}^j match the constraints of s_1 and symbol recognition system recognizes s_1 before s_2. Based the definition of SPR, SPR(s_1)=SPR(S_{12})+SPR(s_{11}) and SPR(s_2)=SPR(s_{12})+SPR(s_{22}). If SPR(s_1)>SPR(s_2) then SPR(s_{11})>SPR(s_{22}). If SPR(s_{11})>SPR(s_{22}), it will be under two cases: (a) complexity of s_{11} is high than s_{22}, (b) There are more similar objects to s_{22} than to s_{11} in the drawing. In case (a), s_1 will have more constraints than s_2. In case (b), the probability of related errors of s_2 will be higher than s_1. Therefore in both cases error rate of condition (2) will be lower than that of condition (1). So $p(s_1s_2) \geq p(s_2s_1)$.

If $\{s_1, s_2\} \subset R_3$, then $p(s_1 s_2)$ will be equal to $p(s_s s_1)$.

Therefore, for a symbol set, to change the recognition order based on the symbol recognition priority will avoid some related errors, so that the total recognition accuracy can be improved.

The proof of efficiency improvement is given below. If there are two symbols s_1 and s_2. s_1 consists of a_1 lines and c_1 strings, s_2 consists of a_2 lines and b_2 edges and numbers of to be lines, edges and strings in drawings are A, B and C, respectively Two factors will affect SPR: the complexity of the symbol model and recognized primitives in drawings. In most cases SPR is in inverse proportion to the number of recognized primitives in drawings. If $A > B > C$, $SPR(s_1)$ will be higher than $SPR(s_2)$ under most conditions. The efficiency will be improved if the symbol recognized system recognizes s_1 before s_2.

The average searching cost of selecting one graphic primitive from M graphic primitives is $\frac{M}{2}$. So the average searching cost of selecting M graphic primitives from N graphic primitives is $\frac{M!}{2^N (M-N)!}$. If the system recognizes S_1 firstly, and then recognizes S_2, the matching cost QS_1 will be:

$$QS_1 = \frac{A!}{2^{a_1}(A-a_1)!} \bullet \frac{C!}{2^{c_1}(C-c_1)!} + \frac{(A-a_1)!}{2^{a_2}(A-a_1-a_2)!} \bullet \frac{B!}{2^{b_2}(B-b_2)!}$$

If the system recognizes S_2 first, and then recognizes S_1, the matching cost QS_2 will be:

$$QS_2 = \frac{A!}{2^{a_2}(A-a_2)!} \bullet \frac{B!}{2^{b_2}(B-b_2)!} + \frac{(A-a_2)!}{2^{a_2}(A-a_1-a_2)!} \bullet \frac{C!}{2^{c_1}(C-c_1)!}$$

Then the value of QS_2 / QS_1 will be:

$$\frac{B!}{2^{a_2+b_2}} \bullet \left(\frac{A!}{(A-a_2)!} - \frac{(A-a_1)!}{(A-a_1-a_2)!}\right) / \frac{C!}{2^{a_1+c_1}} \bullet \left(\frac{A!}{(A-a_1)!} - \frac{(A-a_2)!}{(A-a_1-a_2)}\right)$$

Based on statistical data, in a symbol the number of any kind of graphic primitive, except line, is no more than 4. For $a_1 > 0$, $c_1 > 0$, $a_2 > 0$, $b_2 > 0$, so $a_1 + c_1 - a_2 - b_2 \in [-4,4]$. It means that $2^{a_1+c_1-a_2-b_2} \in [\frac{1}{16},16]$. In an engineering drawing the total number of one kind of graphic primitive will be more than 100. If the drawing is slightly complex, the number will be over 1000. In most conditions, $B - C > 5$, so the value of

$$\frac{B!}{2^{a_2+b_2}} / \frac{C!}{2^{a_1+c_1}} = \frac{B! \bullet 2^{a_1+c_1-a_2-b_2}}{C!}$$ will be more than 100000. For

$$\left(\frac{A}{(A-a_2)!} - \frac{(A-a_1)!}{(A-a_1-a_2)!}\right) / \left(\frac{A}{(A-a_1)!} - \frac{(A-a_2)!}{(A-a_1-a_2)}\right) = \frac{(A-a_1)!}{(A-a_2)!}$$

If $a_2 > a_1$ and $a_2 = a_1$, it is so obvious that the $QS_2 / QS_1 >> 1$. If $a_2 < a_1$, in a symbol the average difference value of a_1 and a_2 will be no more than 2. So $\frac{(A-a_1)!}{(A-a_2)!} \approx \frac{1}{A-a_2}$, it is also obvious that $QS_2 / QS_1 >> 1$. This proves that by only changing the recognition order, the recognition efficiency will be improved greatly.

4.4 Experiments

It is common for a symbol to be a subset of another symbol, so related errors exist in different applications of symbol recognition. Experiment one includes a survey on the subset-relationship that exists in the symbol models presented in the previous studies. Five types of drawings are selected, they are: maps, geographic drawings, electronic drawings, circuit diagrams and constructional drawings, and results are shown in Table

4.3. From the survey, subset-relationship is more common in electronic drawings, circuit diagrams and constructional drawings than in the other two types of drawings.

Table 4.3 Symbol Relationship Survey Results

Field	*Symbol Number*	R_1	R_2	*References*
Maps	6	33.3%	0	E. Reiher et al.1996
Geographic	18	5.5%	72.2%	H. Samet et al. 1994
Electronic	25	20.0%	80.0%	A. D. Ventura 1994
Circuit Diagram	8	25.0%	75.0%	A.Okazaki. 1988
Construction	55	23.4%	76.6%	S. Wang et al. 2002

Three kinds of matching orders were chosen and applied in a graph-based symbol recognition system: Random order, high-priority-first-recognition order and low-priority-first- recognition order (see Fig. 4.3). The symbol recognition system is based on attributed relationship graph matching. Most of the traditional systems are currently using the random order. In the High-priority-first-recognition order, symbols with higher SRP values will be recognized before those with lower SPR values, while in the low-priority-first- recognition order, the recognition will be performed in the reverse order to the High-priority-first-recognition order. Low-priority-first- recognition order is used to simulate the worst case in matching order. 15 symbol models and 13 engineering drawings were selected for testing. The largest drawing has 5839 lines, 873 texts and 81 edges, and the smallest drawing has 1335 lines, 582 texts and 21 edges. The results are shown in Fig. 2 and average correct recognition rates of three orders are 96.16%, 93.16 and 89.16%, respectively. By adopting high-priority-first-recognition order, the recognition system can improve correct recognition rate by 3 percent compared with the average.

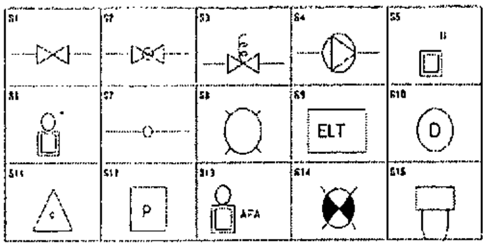


Fig. 4.3 Candidate Symbol Set

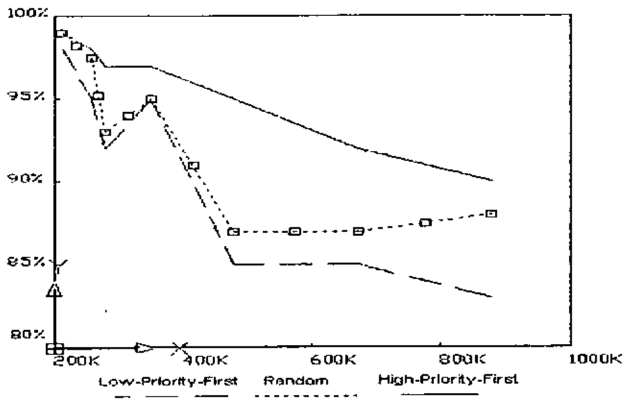


Fig. 4.4 Comparison of Three Kinds of Order Symbol Recognition Order

4.5 Conclusions

A symbol recognition order-adjusting algorithm is presented in this chapter. The method prioritizes the symbols to be recognized by using the similarity measurement of instances in a drawing under two recognition thresholds. Through counting the number between two thresholds, the probability of error recognition of the symbol can be determined. Based on the error probability sequence, a suitable recognition sequence is performed and some of the related errors can be avoided, so that both the accuracy and efficiency of the symbol recognition system can be improved. The algorithm is general and has a broad applicability. Although the example is drawn from symbol recognition to illustrate the point, the method described here can also be applied to other fields of graphics recognition such as OCR.

Chapter 5

A NOVEL WALL DETECTION ALGORITHM BASED ON DOOR SYMBOL RECOGNITION IN ARCHITECTURAL DRAWINGS

5.1 Introduction

Walls are backbones of architectural drawings. It is a basic requirement of quantity survey systems to extract wall information. The quantity of none structural walls could only be recognized from architectural drawings, and wall information in architectural drawings describes the essential spatial information and dimensional information of a building that is important for the quantity survey of structural elements. In the past two decades there has been a growing interest amongst researchers in studying the field of architectural drawings understanding (Tombre 1995; Lladós et al. 1999; Ah-Soon 1998; Pollock 1994). Several methods are potentially useful for extracting graphical information of walls from architectural drawings. These methods could be divided into two classes: semi-automatic extracting (Zhi et al. 2003) and automatic extracting methods (Ah-Soon et al. 1997; Kernighan et al. 1996; Kwon 2003; Sánchez et al. 2003). Semi-automatic extracting methods mainly use layer and block settings to classify and group graphical objects in CAD systems. The algorithms for wall extracting are only performed on the grouped objects. Semi-automatic methods have advantages when drawings are correctly set and prepared. However, current CAD drawings are not produced with standardized settings and symbols. In addition, mistakes, such as wrong layer and object definitions, incorrect placing of objects in layers, redundant objects existing in different layers, are common in current professional architectural drawings. The fact that architectural drawings are prepared in different companies and organizations with different drawing styles and settings adds to the complexity and heterogeneity of architectural drawings. As it is very difficult to require architects to follow the same layer and blocking settings, the semi-automatic extracting methods cannot be widely applied.

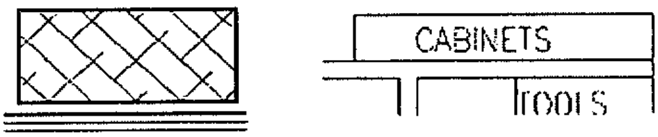
Automatic extracting methods process all the graphical objects and recognize walls by using features. Parallel lines of a wall are a feature applied by most automatic extracting methods (Ah-Soon et al. 1997; Anchez et al. 2003). The key idea introduced by Ah-Soon and Karl Tombre (1997) is to sweep pairs of parallel thick lines that belong to the same wall. However, one drawback is that it is difficult for the sweep algorithm to recognize an open-air courtyard in the drawing. Although Anchez (2003) reported an algorithm which improved Ah-Soon and Karl Tombre's algorithm (1997) by adding some constraints, the additional constraints are still based on the parallel feature of wall. The first attempt of using other helpful information for wall extraction is introduced by Kwon (2003). He presents a method for recognizing main walls in architectural drawings by using dimension extension lines. The main limitation of this method is that a dimension extension line should be the only auxiliary line in wall boundaries. Thus this method cannot recognize and extract walls that comprise several parallel lines.

When the parallel feature is not distinct enough for wall extraction, all the previous automatic methods will have limitations in handling representations of walls and the other architectural elements similar to walls. These limitations are difficult to solve by only adding additional constraints into existing automatic methods. A possible solution is to introduce more features in wall recognition and extraction, but wall representations are simple, features such as dimension extension lines inside wall boundaries can only aid some cases. In this chapter, an automatic wall extraction algorithm that utilizes doors as an additional feature is introduced. The algorithm goes like this. Firstly, a door is recognized for guiding room boundary searching. Then, separating walls between two rooms can be identified. Lastly, the separating walls guide the recognition of other walls that are not between the two rooms. Some examples are used to indicate how this new algorithm overcomes the shortcomings of the previous algorithms and that the speed and accuracy of the presented approach are improved significantly.

5.2 Wall Detection Algorithm Based on Door Symbol Recognition

5.2.1 Difficulties of Wall Recognition

Wall is a continuous structure forming one of the sides of a building or room. It is usually represented with two or several parallel lines. So the parallel feature seems to be the only distinct feature for wall extraction, and is also widely used in many existing algorithms.



a b

Fig. 5.1 Elements similar to wall

Many other elements in engineering drawing have a parallel feature. It is not easy to distinguish a wall from those similar elements (see Fig. 5.1). In addition, as several parallel lines can be used to represent a wall, in some cases all the lines are wall boundaries, while in other cases only the two outside lines are wall boundaries and the other lines are auxiliary lines. Therefore, it is difficult to detect the correct wall boundary by only applying parallel feature (see Fig. 5.2).

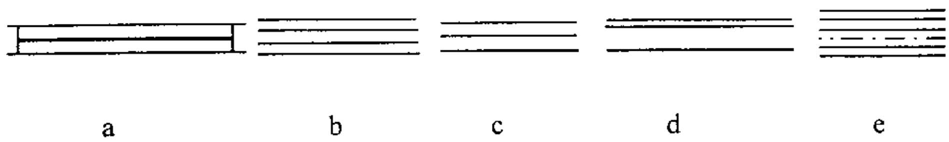


Fig. 5.2 Walls presented in several parallel lines

It is difficult to solve the above problems by only improving the wall-extracting algorithm itself. This is why doors are introduced as an additional feature to guide the extraction of wall boundaries. A door is an essential element opened in the wall, therefore the lines connected with a door symbol must be wall boundaries or auxiliary lines inside them. The representation of door symbol is very stable, and is usually represented with two straight perpendicular lines and an arc, which is much more distinct than the representation of a wall.

5.2.2 Door Symbol Recognition

Symbol recognition is the task of finding and labeling part of a 2D model in engineering drawing. Many algorithms, such as model matching, graph matching, syntactic/structure matching and so on, are proposed for symbol recognition. In this chapter, a rule-based algorithm is adopted for interpreting the door symbol. The main rules are described as follows:

Rule 1: There are two kinds of door symbols: Double Swings Door, Single Swing Door (see Fig. 5.3)

Rule 2: A standard Double Swings Door symbol contains five elements: one Long Door Edge Line, two Short Door Edge Lines and two Door Track Arcs.

Double Swings Door Single Swing Door

Fig. 5.3 Door Symbol

Rule 3: A Long Door Edge Line is the line that shows the width of the door and it intersects with Short Door Edge Lines and two Door Track Arcs.

Rule 4: A Short Door Edge Line is the line that shows the width of door from another direction. The length of Short Door Edge is half of Long Door Edge.

Rule 5: A Door Track Arc is the arc that shows 90 degrees rotation track from Short Door Edge to Long Door Edge.

Rule 6: A standard Single Swing Door symbol contains three elements: two Short Door Edge Lines and one Door Track Arc.

Rule 7: Two Short Door Edge Lines show the width of door in different directions.

It should be noted that the algorithm of door symbol recognition is not the focus of this chapter, and its details could be found in chapter 4 and 7 and a previous paper (Wang et. al. 2002).

5.2.3 Wall Extraction Algorithm

Before wall extraction, the architectural drawing is firstly converted into a graph structure. In the structure, all the intersection points of lines and arcs are the nodes, and lines and arcs are the edges. Next all the recognized door symbols are converted to edges in the graph structure, called door edges (Zhi et al. 2003). Then wall extraction can be performed on the graph structure. There are four steps for wall extraction:

Step 1: Door edges are used as seed segments for tracing the inner boundaries of rooms (IBR), which are also the boundaries of walls.

Step 2: Separating walls between two rooms are detected by analyzing the relationships between rooms. Then the edges in the IBR as the boundaries of separating walls are marked. The left unmarked edges in the IBR are used to find outside walls in step 3.

Step 3: Outside walls are not between rooms. They are used for surrounding rooms or buildings. The objects adjacent to outside walls might be vacancies or other objects such as staircase, corridor and water tanks. These walls are recognized under the guidance of the unmarked edges and the recognized separating walls.

Step 4: Junctions between walls are detected to connect all the walls.

The details of these steps are described in the following sections.

5.2.3.1 Inner Boundaries of Room Detection

An IBR is a set of edges in the graph structure surrounding one room. The detection algorithm includes the following step:

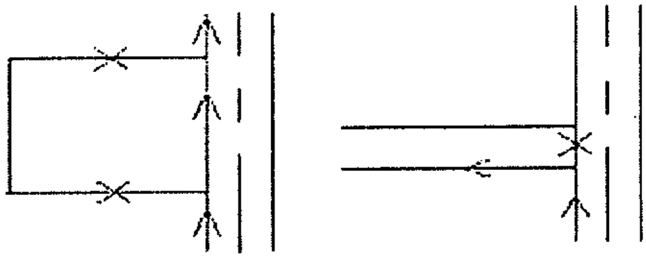
Step 1: Get one door edge (Vs,,Ve) overlapped with the inner side of room, Vs is the start point and Ve is the end point. Put the edge (Vs,Ve) into the a new empty set St. Take this edge as the seed segment to track the other connected edges. Set the segment edge E= (Vs,,Ve)

Step 2: Get all the edges (V_k, V_{k+1}) (j=1 n) connected with E at point Vs.

Step 3: Find the first edge (V_k, V_{k+1}) $(1 \le k \le n)$ that fulfill both of the following conditions

(a) There exists an edge in the graph that is parallel to (V_k, V_{k+1}), and the distance between them is less than the maximum thickness of wall (Fig.5.4a).

(b) The direction from E to edge (V_k, V_{k+1}) is anti-clockwise(Fig.5.4b).



a b

Fig. 5.4 Edge Tracking Examples

Step 4: If the edge (Vk ,Vk+1) is equal to (Vs,,Ve), the IBR detection for door edge (Vs,,Ve) is finished. Otherwise put (V_k, V_{k+1}) into St and set $E=(V_{k+1}, V_k)$, then go to step 2.

Step 5: Select another door edge as seed segment, and go to step 1. If there is no unselected door edge, go to step 6.

Step 6: Delete the duplicated St ($1 \leq t \leq m$) that contains the same edges. The left St are the inner boundaries of rooms (IBR) in the drawing.

Fig. 5.5 gives an example of IBR extraction. There are two rooms in the sample: NIGHT REST RM and LAV. First the two door edges are (V1, V9) and (V10, V15) recognized based on the door symbol recognition, then the IBRs are analyzed from points V9 and V15, finally the IBRs of two rooms {V1, V2, V3, V4, V5, V6, V7, V8, V9} and {V10, V11, V12, V13, V14, V15} are extracted.

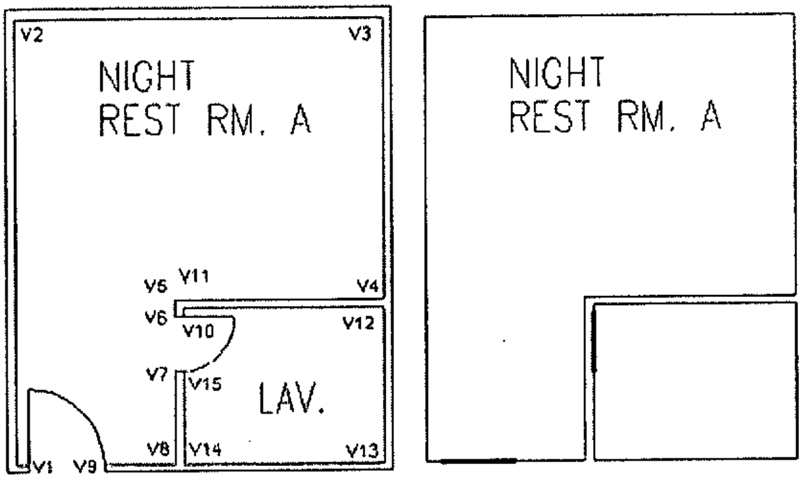


Fig. 5.5 Wall Boundary to Inner Room Boundary

5.2.3.2 Separated and Outside Wall Extraction

After establishing the IBR, the separating wall can be identified easily. The relationships between two rooms can be classified into four types: (a) adjacent with one edge, (b) adjacent with several edges, (c) intersection at a corner, (d) separate. In the conditions of type a and b, the edges between two rooms in IBR can be recognized as the separating walls. Then the outside walls can be detected based on the rest edges in IBR. The wall detection algorithm is described in detail as follows (see Fig.5.6):

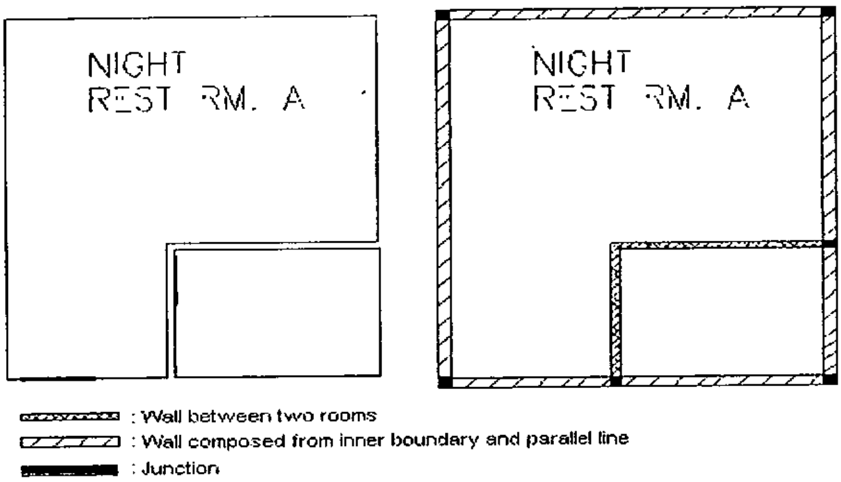


Fig.5.6 From Room to Wall

Step 1: Check every two edges (Vi, Vj) and (Vp, Vq) in different rooms. If the two edges are parallel and their distance is less than the maximum thickness of wall, (Vi, Vj) and (Vp, Vq) are the boundaries of a separating wall.

Step 2: Make projection between the two edges. There are four projection points in the edges. The points not overlapped with the two edges are deleted.

Step 3: Using these points, each edge will be divided into several segments (at most three segments). Replace the original edges with new separated edges in IBR. Select the two divided edges overlapped in the projection direction as the boundaries of a separate wall, which are marked for the following use.

Step 4: After extracting all the separate walls, select each edge of every room that is not marked. Get the thickness of the nearest separating wall which connects to the edge. Then find its parallel lines where the distance between them is the same as the above thickness. These lines and the edges comprise the outside walls.

5.2.3.3 Wall Connection

Until now all the walls have been detected. The junctions connecting the extracted walls could be extracted at this time. Every two recognized walls intersected at a point are established, and the lines connecting these two walls are then found. There are four kinds of junctions: +_Junction, T_junction, L_junction and Δ_junction. (see Fig.5.7). The

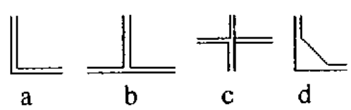


Fig. 5.7 Junctions

junction comprising the lines and wall boundaries are generated based on the intersecting positions of these lines. Finally the junctions are used to connect the walls.

5.3 Experiments

In this section, two representative experiments are reported in the context of wall recognition in architectural floor plan and interior decoration plan. In the experiment one the presented wall extraction method is compared with those of Karl Tombre (1997) and Kwon (2003). The architectural floor plan (Fig. 5.8) is chosen for the comparison. In the experimental plan, there are 22 rooms, 39 doors and four types of wall: 127 Single Walls, 26 Three Lines Walls, 15 Glass Walls and 7 Six Lines Walls (see Fig.5.8). The recognition results are displayed in Table 5.1.

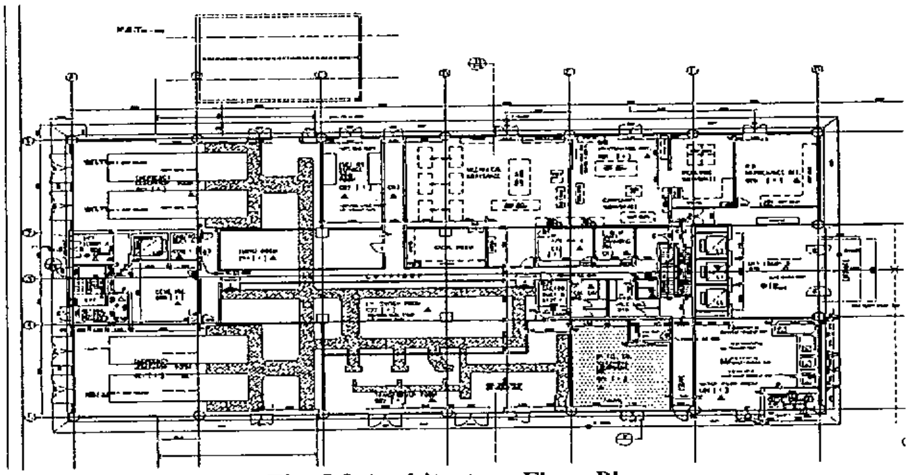


Fig. 5.8 Architecture Floor Plan

Table 5.1: Results of Wall Extraction

Wall Type													*Rec. Rate*
	R	E	M	R	E	M	R	E	M	R	E	M	
Tombre	111	12	16	0	0	0	0	0	0	0	0	0	87.4%
Yong-Bin	114	12	13	0	0	0	0	0	0	0	0	0	89.7%
Proposed	124	0	3	25	0	1	15	0	0	7	0	0	97.6%

R: Successfully recognition, E: Error Recognition, M: Miss Recognition

In experiment two an interior decoration plan is chosen for wall extraction (see Fig. 5.9). The interior decoration plan is based on architectural plan. Some appending information such as furniture, decoration material and ceiling are also drawn on the drawing. In such a plan, some elements are very similar to a wall, and give many difficulties in wall extraction. In Fig.5.9, there are 16 rooms, 68 single walls, and 12 three lines walls in the plan. The recognition results are displayed in Table 5.2.

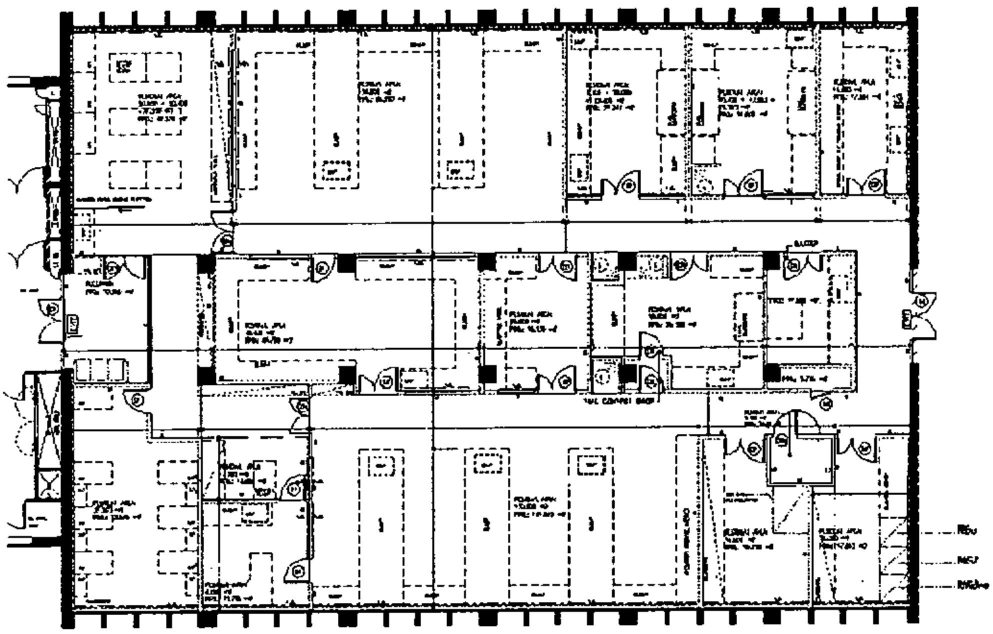


Fig. 5.9 Interior Decoration Plan

Table 5.2: Result of Wall Recognition in Interior Decoration Plan

Wall Type							*Rec. Rate*
	R	E	M	R	E	M	
Tombre	61	12	7	0	0	0	89.7%
Yong-Bin	61	12	7	0	0	0	89.7%
Proposed	68	0	0	12	0	1	98.8%

R: Successfully recognition, E: Error Recognition, M: Miss Recognition

In experiment two, methods of Tombre and Yong-Bin's have the same recognition rate because there is no dimension extension lines in the interior decoration plan. Though the recognition rate of their methods increase in the experiment two, the number of error-recognized walls increases too. The reason is that their methods could not distinguish the elements similar to a wall from the real walls.

5.4 Conclusions

This article presents an automatic method to extract wall information from architectural drawings. Most of the methods that have been applied in the past have come across some difficulties in processing such complicated drawings, and could not achieve high recognition accuracy. Some systems use a semi-automatic method to get the wall and room information by setting layers and blocks. However, the efficiency and accuracy are not very satisfactory. There are two reasons that dominate the accuracy of wall recognition: different representations of a wall and architectural elements similar to wall. The presented method uses the door as the seed segment to guide the recognition of inner room boundary. Firstly, separating walls between rooms can be extracted. Then outside walls are recognized with the help of the separating walls. Finally all the extracted walls are connected with junctions. In the previous methods every line in a drawing need to be swept which is time-consuming. In the presented method only the lines that are possible walls are processed, so the accuracy and efficiency of the presented method are improved significantly.

The method presented in this chapter aims at wall recognition. However, the idea that an object with high recognition accuracy may be used to guide the recognition of the objects with low recognition accuracy has a broader applicability. Although an example is drawn from wall recognition and illustrate the point, the method described here can also be adopted in other fields of graphics recognition such as drawing Vectorization.

Chapter 6

AUTOMATIC ACQUISITION OF RULES FOR UNDERSTANDING AND INTERPRETATION ENGINEERING DRAWINGS

6.1 Introduction

In the design of the EDUI system, the issue of knowledge acquisition, organization and contextual reasoning is crucial. The target of the research is the conversion of engineering drawings in the construction industry to high-level representations with sufficient information for later stage analysis. However, the aim is not only to build a universal system capable of understanding and interpreting engineering drawing in the construction industry, but also to explore the power of knowledge-based techniques for performing high-level understanding and interpretation of engineering drawings in other industry.

Similar to many other existing technologies, knowledge is first converted into recognition rules in the understanding and interpretation system manually. Though this kind of static-rule-based methods can achieve good results, it also brings some drawbacks. Firstly, manual rule analysis is time-consuming and the adaptability of the recognition algorithm isn't good (Prabhu et al. 1999; Rangachar et al. 1990; Ah-Soon 1998; Llados 1998; Yu 1997). Secondly, discussions are usually limited to symbol recognition. But for those complex graphic objects, the recognition ability is lower (Ah-Soon 1998; Llados 1998; Yu 1997). Thirdly, those methods based on object feature template are too closely related with the structure definition of templates. The recognition algorithm must be modified with the change of the template structure and the flexibility is low (Prabhu et al. 1999; Ah-Soon 1998; Yu 1997).

Some researchers have proposed methods for automatic rule learning for drawing recognition (Samet et al. 1994; Satoh and Mo et al. 1993). However, only a few of them have actually implemented the learning procedure. The decision tree is one of approaches. The values of one simple primitive attribute or dualistic relation (such as inclination) are collected through strict training of many samples. These values are the only basis for choosing the cut points of the decision tree (Lu, Wu, Sakauchi 1995; Satoh, Satou, Sakauchi 1992). In addition, the user is allowed to decide which branches are more important and which can be deleted (Kefalea et al. 1999). Thus, the tree structures are always very complex, lacking optimized path arrangement, and long traversal time is needed for recognition. Generally, they are only suitable for very simple graphics objects, such as "two cross lines", "triangle" etc.

The other class is devised for classifying non-self-intersecting shapes that comprise straight lines only. It analyzes conjunctions of local properties of each shape, indexes all the shapes by the properties and matches the indexed shapes against the instance by calculating the sum of the weights of successfully matched properties. Weights are manually selected and modified until all the samples can be recognized correctly. This approach needs a lot of samples and several different training sequences for one object type. Furthermore, it cannot adapt to objects with complex graphics (self-intersecting, or containing arcs, circles, strings) that appear frequently in architectural drawings.

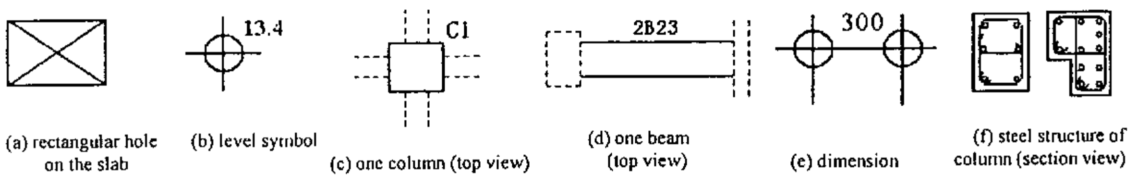
To solve above problems, this chapter presents a new heuristic sample-based rule acquisition method for self-adapting recognition system. This method introduces the combination of *single-sample-analysis* and *multi-sample-comparison*. Based on the integrated calculation of primitive attributes and relationships, supported by heuristic principles derived according to general knowledge, feature extraction, reduction, synthesis and comparison are performed. Recognition rules are then converted from the features. Similar to a human's learning procedure, while more new object representations and types are being processed, rules are perfected step by step and the recognition effect improves. This method can deal with complex objects and has no special requirements on the sample or training sequence.

6.2 Self-adapting Recognition Mechanism

Graphic primitives (GP) in engineering drawings can be classified into two groups: lines (points, straight lines, circles and arcs) and strings. One group of GPs, which has certain domain meaning, is called "*Architectural Object*" (for short, the term *object* has been used in this chapter), such as "level symbol", "column", "beam", "wall" etc.

6.2.1 Object Graphics and Corresponding Recognition Method

Different objects have different graphics representation characters, thus corresponding



Note: Dashed lines represent the GPs that don't belong to but are connected with the object graphics. They may not be dashed in real drawings.

Fig. 6.1 Examples of Architectural Objects

recognition methods are different too. Several typical examples are given in Fig. 6.1.

(1) Some objects' graphics have fixed topology and the topology matching is used to recognize them. Fig. 6.1(a) shows one rectangular hole on the slab. Its size and aspect ratio change greatly according to the reality, but it has the fixed topology: "one rectangle contains two catercorner lines intersecting each other". Such kind of objects can therefore be recognized through matching the topology structure.

(2) Some objects' graphics contain one special GP fit for leading the recognition. Fig. 6.1(b) shows one level symbol, composed of one circle, two perpendicular lines and one string. Because the circle is usually a small part of all the GPs in a whole drawing, for improving the recognition efficiency, the circle in this symbol should be recognized firstly.

(3) Some object's graphics contain one string, which has particular composition rule and so can be the lead of the recognition. Solid lines in Fig. 6.1(c) compose one rectangular column's boundary in the top view and dashed lines belong to the beams that are possibly connected with this column. String "C1 is column name, in which 'C' is the prefix, followed by one serial number. This is one of the naming rules of the column in engineering drawings in Hong Kong construction industry. String's amount is also relatively small in one drawing and content-based string matching is fast, so when recognizing such kind of objects, for improving the efficiency, search the special string to locate the object firstly, then search other lines beside the string.

(4) Some object's graphics contain certain strings, but their contents are not regular, so the lines must be recognized first and then the existence of needed strings examined. In Fig. 6.1(b), '13.4 is the string denoting the level value of the symbol. This string

must exist but the value cannot be, so it will be searched after the lines are recognized.

(5) Some objects have particular connected relations with environmental lines. These relations must be validated after recognizing the object graphics. Dashed lines in Fig. 6.1 (c) show a kind of unfixed relation because their quantity and position of are changeable. As for Fig. 6.1 (d), dashed lines belong to two objects that are connected with the beam (named "2B23) by one line respectively. One of these two objects must be the construction component such as column, beam or wall, the other one may be the construction component too or the "beam border" that is just one line indicating the end of beam. This is a kind of fixed relation.

(6) Exclusive checking is needed after the recognition of one object, if it is not permitted to have any connected relations with other objects or its graphics is a part of another one object's. Fig. 6.1(e) shows one dimension. It can be looked as two level symbols, one of which there's no string. If one level symbol is found, additional examination should be undertaken to confirm that no other level symbol (without string) existing beside found one, viz it isn't a dimension.

(7) Some objects' graphics contain GPs that can be classified into several groups. GPs of each group compose one kind of pattern whose quantity is variable and the arrangement of the groups is regular. The recognition method is pattern matching aided by regularity checking. Fig. 6.1(f) shows a steel bar structure in section view of two columns (strings are removed). One polygon of the outer layer represents the column boundary, within which the steel bars are represented by smaller polygons, short lines and small circles cling to them. This regularity then is used for

recognizing such kind of objects though the quantity and position of these steel bars will change greatly with the column's size and shape.

6.2.2 Self-adapting Recognition Mechanism

The Graphics representation characters of objects determine their recognition methods. Therefore, before concluding the method of recognizing one type of objects, all the possible representations and their features must be analyzed. But objects of the same type may have totally different shapes if different drafting conventions are applied or even only the draftsmen are different, so it's impossible to forecast what kinds of representations will appear. Ordinary rule-based recognition approaches always need to modify the predefined object templates or recognition algorithms manually when meeting new representations of objects. This situation reduces the practicability of the approaches. This chapter presents a brand-new "self-adapting" recognition mechanism. In this mechanism, the function of object graphics analysis and rule definition, previously done manually, is moved into the recognition system. The recognition rules for any type of objects will be automatically generated and modified to adapt new

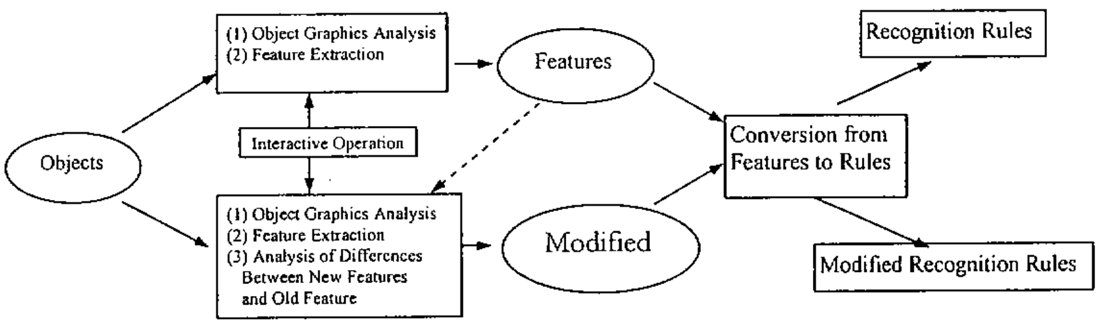


Fig. 6.2 Automatic Generation and modification of Object Recognition Rules

appearances of objects, without changing the algorithm. Fig. 6.2 shows the module that implements the generation and modification of the rules.

For the given objects of type T, when users designate the first sample of T, the system analyzes it to extract out the graphics features. These features are confirmed or partially modified by users according to their experiences, and then they are reserved as the representative features of T. Meanwhile, rules for recognizing T are automatically converted from the features. When only one sample of T is processed, even if the user has made some modifications, the adaptability of obtained rules is comparatively weak because they will fail to recognize those objects belonging to T but have totally different shapes from the first sample. Therefore, when users designate another sample of T, the system extracts features from the new sample, and then compares new features with old ones. Through analyzing the differences, the system can automatically (interactive operation can be done if needed) modifies the features and generates new rules that have better adaptability.

Along with the introduction of more different object representations or types, the system continuously modifies and improves the recognition rules, the recognition module can then use new rules to achieve better recognition effect. That is, the system's adaptability will be improved without modifying the program.

6.3 Object Analysis and Feature Extraction

Object features embody a group of stable relations among the GPs composing this object. Accordingly, recognition rules are the criteria to judge whether the relations in object features can be satisfied. Automatic feature extraction is therefore indispensable for rule acquisition.

6.3.1 Designation of Object Graphics

Given one sample drawing of one object type, users designate the object's graphics through following three steps:

(1) Draw one rectangular encircling the whole object graphics and give the object type.

(2) In the box, designate NI: the **strings and lines** that don't belong to the object and are irrelevant with the object.

(3) In the box, designate NC: the **lines** that don't belong to the object but have certain connected relations with the object.

Then the left GPs in the box belong to the object graphics.

6.3.2 Feature String and Description of Its Composition Rule

Many types of objects have their own "names". In the graphics, the object name is just one string, which has one or several kinds of fixed composition rules. Because these strings directly indicate the location of corresponding objects and make the recognition much more effective, they are defined as *Feature String* (FS). FS's composition rules are related to drafting conventions and cannot be analyzed automatically, so this chapter presents a method for users to describe it conveniently. Several characters that will not appear in real engineering drawings are chosen for the purpose of format control. For instances, '?' denotes one integer (one digit or long number), '#' denotes any continuous 'blank' character, '^n' denotes any non-null string whose length is less than n (n is an integer), each part bracketed with one pair of '{' and '}' is an omissible part.

For example, in engineering drawings used in the Hong Kong construction industry, prefixes of beam names are always floor number followed by 'B' and size annotation can also be included, such as "1B1 "5B22 "1B1 (400X600)" etc, so "?B?#{(?X?)}" is defined to describe the composition rule of this sort of beam names.

6.3.3 Closed Line Series

In object graphics, there're always one or more *Closed Line Series* (CLSs), each of which consists one circle or several lines (arcs) connecting end to end. One of them possibly can be the representative feature depicting the object's shape. During recognition, CLSs benefit locating the object quickly, especially when it contains the circle, arcs or regular polygon such as square and equiangular triangle.

6.3.4 Object Graphics Analysis

Considering the characters of architectural objects, analyze the designated object graphics by following steps:

(1) Separate strings from lines. Figure out two minimal boxes, "*Line Box*" is the one encircling all the lines, "*Object Box*" is the other encircling all the strings and lines.

(2) Analyze the relation between NI, NC and object graphics. If there is no GP of NI within Line Box, then assumption is made that this object excludes other GPs' existence. If some lines of NI are connected with object's graphics on their

endpoints (such as dashed lines in Fig. 1,c), then this object is considered to have flexible connected relations with environmental graphics. If NC isn't null, then this object is considered to have fixed connected relation with the lines of NC.

(3) Analyze the features of lines:

- Search out all the CLSs.
- If one or more CLSs exist, begin from the largest one, check the filling pattern of each CLS: whether it is filled with one kind of hatch or several groups of GPs each of which has one certain pattern (see also Fig. 1, f).
- Arrange all the CLSs in following sequence: circle, polygon including arc(s), square, equiangular triangle, rectangular, ordinary polygon. The first one is defined as *Feature Line Series* (FLS).
- If FLS exists, then analyze the relation between it and other CLSs or other individual lines.
- If there's no FLS, well then there's no CLS, then define the longest arc (if existed) or line as *Feature GP* (FGP) and analyze the relation between FGP and other GPs.

(4) Analyze the features of strings:

Search for one string according with the composition rule of FS of the object. If successful, extract the string out as *Leading String*, then search for the down-leads, each of which points at the *Leading String* and object lines respectively with two ends of it.

For other strings, analyze the relation between *Leading String* and them, as well as the relation between FLS or FGP and them, and then search for the down-leads pointing at these strings and object lines.

6.3.5 Extraction of Object Feature

Feature Groups

After analyzing the object graphics, the result is classified into six feature groups, which together describe the integrated features of each type of objects:

(1) *Leading Feature.* This defines that which part of the object graphics should be recognized firstly and how to recognize this part, comprising four aspects:

1. Leading part's type: *Leading String*, FLS or FGP.
2. Leading part's information: descriptions of *Leading String*'s composition rules, geometric structure of FLS or geometric attributes of FGP.
3. Matching constraint: indicates how to match the leading part (if it is FLS or FGP) with the real drawings during recognition, embodying *Topology Matching*, *Direct Matching* that means the graphics of leading part must be matched without changing the shape, size or rotation, *Scaling Matching* that means the size can be zoomed when matching, *Rotated Matching* that means rotation is allowed when matching.

4. Leading way: indicates the relationship between the leading part and other parts of object graphics, such as down-lead leading, position leading and distance leading.

(2) *Closure Feature.* It defines the characteristics of all the CLSs and how to recognize them, comprising four aspects:

i) Structure feature: geometric structure and size of each CLS.

ii) Filling feature: indicates each CLS's filling pattern, which is described in. If one CLS is filled with nothing then its filling feature is null.

iii) Matching constraint: indicates how each CLS should be matched. Same as that of *Leading Feature,* it embodies *Topology Matching, Direct Matching, Scaling Matching* and *Rotated Matching.*

iv) Leading GP: defined as the longest arc for each CLS including arc(s) and the longest line for others.

(3) *Relation Feature.* It describes all the relations among the GPs of the object graphics.

i) Relating way: similar with "leading way", indicates how some GPs are related with some others, embodying down-lead relating, distance relating, position relating and enclosure relating.

ii) Relating constraint: embodies all the detailed geometric constraints such as the relation between *Leading String* and individual lines, the relation between individual lines and CLSs or FGP etc.

(4) *Transformation Feature.* It defines what kinds of geometric transformation can happen to the whole object graphics in different appearances of this object:

i) Transformation type: no transformation, zoom with changeless or changeable aspect ratio of *Object Box*, rotation.

ii) Transformation constraint: embodied the permitted scope of scaling or rotation angle.

(5) *Exclusive Feature.* It indicates what kind of GPs that don't belong to the object cannot appear within *Line Box.*

i) Exclusive type: no exclusive (exclude nothing), *All Exclusive* (exclude anything), *GP Exclusive* (exclude some certain GPs' existence).

ii) Exclusive information: embodies the geometric attributes of the excluded GPs for *GP Exclusive.*

(6) *Connection Feature.* It describes the necessary relations between the object graphics and environmental lines.

i) Connection type: no connection, *Fixed Connection* and *Flexible Connection*.

ii) Connection constraint: embodies connecting position and type for *Fixed Connection*, permitted quantity and geometric attributes of connecting GPs for *Flexible Connection*.

Initial Feature Evaluation

Features that are extracted from only one sample of one object type may not be integrated enough, because it's impossible for the system to automatically dope out other different representations from current one and the initial values of some features are then set automatically by estimation only. For example, if in current sample there is no GP of NI within *Line Box*, the system will temporarily set the *Exclusive Feature* of this object type as "*All Exclusive*". Another example is, the system cannot make decision from one sample whether one type of objects are permitted to be rotated or zoomed, so temporarily set the *Transformation Feature as* "no transformation".

Through automatic initial evaluation like above, features obtained may have some limitations. Users can modify manually according to their experiences, making the features more adaptable to different representations of the same type of objects. Users also can designate more samples to let the system modify the features automatically.

6.3.6 Modification of Object Feature

When given a new sample of one object type whose features have been extracted before, the system compares the old features with new ones that are extracted from the new sample, and automatically (also reminds the users to check) modifies the object features according to the differences identified in the comparison. For instance:

(1) If the old *Exclusive Feature* indicates "*All Exclusive*" but new one is "*No Exclusive*", then exclusive type of this type of objects will be changed to "*No Exclusive*". Similarly, if old *Connection Feature* indicates "no connection" but new one is "*Flexible Connection*", then "*Flexible Connection*" will be the final choice.

(2) If the matching constraint of old Closure Feature indicates that one CLS should use *Direct Matching*, but this CLS appeared in new sample has the same geometric structure but different size with the old one, then the matching constraint of this CLS will be changed to *Rotated Matching*.

(3) If the position relation between one string and some lines in the new sample is different from that of the old one, corresponding relating way will be changed to "distance relating" from "position relating" and calculated distance is used to determine relating constraint.

(4) If new features are totally different from old ones, such as some CLSs appear in old features but don't appear in the new sample, or GP composition is totally changed (see Fig. 6.3. a and b) etc, then two sets of features are both reserved to be coordinate. When any one of those coordinate features is satisfied, the object is recognized successfully.

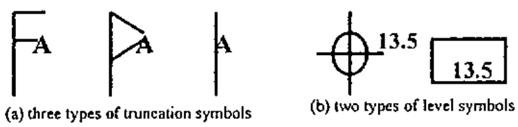


Fig. 6.3 Examples of Feature Modification – Coordinate Features

Because of the diversity of building design, drafting convention and draftsman's custom, when beginning to process the drawings of a new construction project, new object types or new representations of old types will appear. According to the presented method, users can pick out appropriate samples from new drawings in advance, or take out unrecognized objects as new samples; then feature extraction, comparison and modification will be performed automatically; finally the features of new object types are added and the features of old types are modified to be more representative.

6.4 Automatic Generation of Recognition Rules

Whenever object features are newly extracted out or modified, the recognition rules for relevant object types will be automatically regenerated according to the latest contents of features.

6.4.1 Typical Object Recognition Process

For one object possessing the integrated six feature groups explained in Chapter 6.3.5, Fig. 6.4 gives its typical recognition process and shows which features should be used for which recognition phase. If some features are missing, the corresponding phase will be omitted. For example, if the leading part's type is FGP but not *Leading String*, then the first phase "Recognition of Leading String" needn't be done.

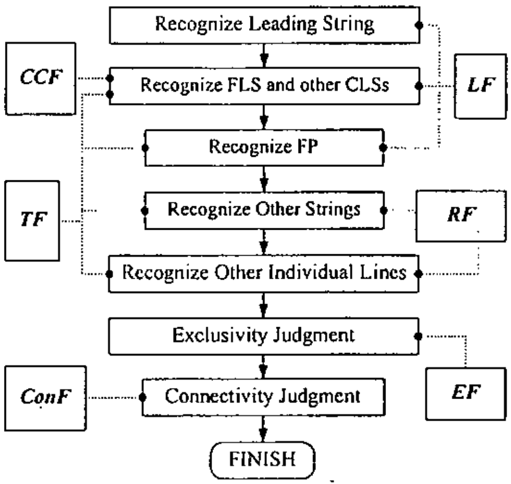


Fig. 6.4 Recognition Phases and Key Feature Group

6.4.2 Conversion from Features to Recognition Rules

It can be concluded from Fig. 6.4 that the recognition process of one object depends on a group of ordered rules, and each rule can be converted from some correlative object features. The following are some examples of the conversion method.

Considering one object type:

(1) Assume that leading type is *Leading String* whose composition rule is predefined as S (for example, S can be "?B?#{(?X?)}"), leading way is down-lead leading: 'LL', then the rule for recognition of *Leading String* is:

IF MATCH_LEAD_STRING(S, 'LL')

THEN RETURN(Names, Leading-lines)

ELSE FAIL

(2) Assume the *Closure Feature* contains following information, one CLS's geometric structure is described as the model: 'L'; filling feature is null: 'NF'; leading GP is the longest line: *a*; matching constraint is *Direct Matching* 'DM'. The *Transformation Feature* indicates that transformation type is "rotation" and permitted rotation angle is from 0 to 360: 'T:R(0-360)', then following rule can be generated for recognizing this CLS:

IF MATCH_LINES('L', 'NF', *a*, 'DM', 'T:R(0-360)')

THEN RETURN(Polygon)

ELSE FAIL

(3) Assume FGP is the longest line: *a*; *Relation Feature* contains the information: there's one string related with *a*, the relating way is distance relating 'DL', relating constraint indicates that the string is parallel ('P') with *a* and the distance between them is less than *d*, then following rule is generated:

IF MATCH_STRING(*a*, 'P;DL:$\leqslant d$')

THEN RETURN(String)

ELSE FAIL

(4) Assume exclusion type is *All Exclusive*: 'ALL', then following rule is used for exclusive judgment:

IF (TEST_EXCLUDE('ALL') $\neq$ NULL)

THEN FAIL

6.5 Experiments

The method applied to the EDUI system that aims at automatic calculation of reinforcement steel in engineering drawings. Fig. 6.5 gives one example of the framing plan recognition in this system.

Fig. 6.5(a) shows a corner of one framing plan (.DWG file), which comprises five types of objects: column, beam, slab symbol, level symbol and truncation symbol.

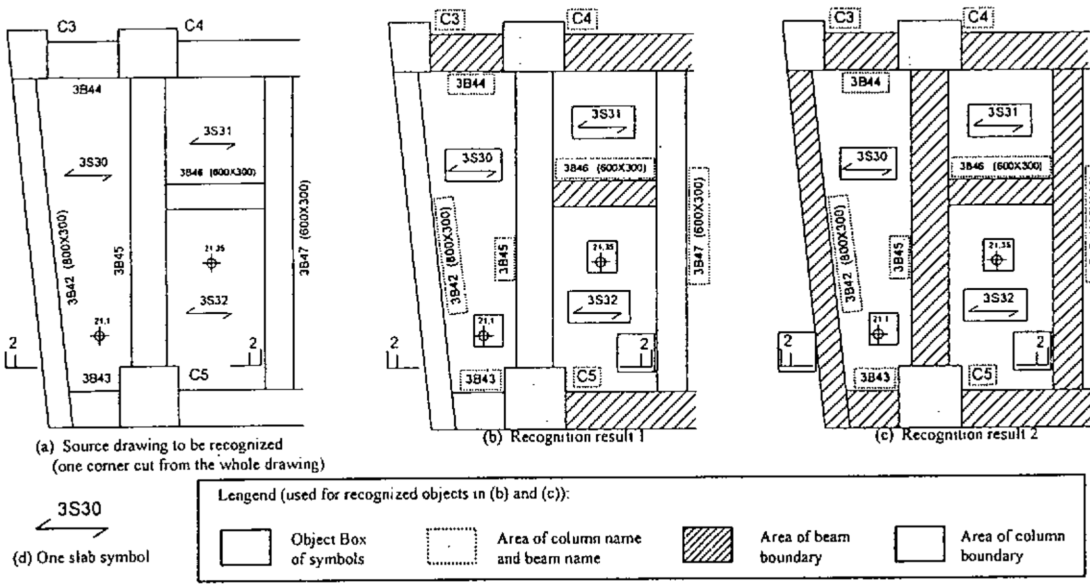


Fig. 6.5 Experiment Result of Self-Adapting Recognition

Predefined FS's composition rules of column, beam and slab symbol are "C?", "?B?#(?X?)" and "?S?" respectively. The system only processes the following samples: one rectangular column, one horizontal beam, one slab symbol, one truncation symbol and one level symbol, which are used to 'train' the system. Though aided by a little interactive modification, the system can self-generate rules based on the features extracted from the samples. However, extracted features have some limitations such as: the *Closure Feature* of a column requires "rectangle", the *Relation Feature* of a beam requires two horizontal lines with the same length; the *Transformation Feature* of a truncation symbol is "no transformation" so *Rotated Matching* is unallowable. According to the rules converted from these parochial features, recognition results are shown in Fig. 6.5(b). Hereinto lines of column C3, lines of beam 3B42(43/45/47) and one truncation symbol are unrecognized.

Users then select new samples from the unrecognized objects, here C3, 3B42 and the one-truncation symbol are chosen. The system extracts features from the new samples and modifies the features of relevant types of objects. When the rules converted from modified features are applied, recognition results are improved as shown in Fig. 6.5(c).

Other similar experiments also manifest that the presented method can effectively improve the system's self-adaptability without modifying the program, therefore the applicability of the system is improved.

6.6 Conclusions

By adopting the function of automatic object feature extraction and rule acquisition presented in this chapter, rules in the recognition systems are continuously refined along with new objects (representations) introduced to the system. The recognition ability is therefore improved without modifying the recognition algorithm. This mechanism brings higher self-adaptability, stability and applicability to the system described in the chapter. Experimental results also reinforce the better performances of the proposed system.

Chapter 7

A KNOWLEDGE BASED ENGINEERING DRAWINGS UNDERSTANDING AND INTERPRETATION SYSTEM FOR AUTOMATIC MEASUREMENT OF STEEL REINFORCEMENT

7.1 Introduction

In the current bidding practice, engineering drawings that are either prepared manually or by using CAD systems, are given to bidders as part of the tender documents. Quick and accurate measurement of the quantities from engineering drawings provided is vital for preparing and winning a bid. In this task, taking off the amount of steel reinforcement is the most time consuming activity, as it requires to manually measure the dimensions and quantities of steel reinforcement in a structure that is projected in three perspectives: plan, elevation and cross section views. For a middle size project, it normally needs 4-5 man-months to take off the quantities of steel reinforcement. This amount of time increases exponentially with the size of project.

Some semi-automatic systems are developed to solve the problem (for example: http://www.sjms.com.cn/sims-cpjs.htm, http://www.tangent.com.cn). These systems provide some models of structural elements and require the users to input the parameters of the predefined models. Quantities of steel reinforcements may then be generated based on the input parameters. But there are several limitations of these systems. Firstly, these systems can hardly generate accurate results because the predefined models are an approximation of actual building objects. So these results are barely useful for cost estimation and could not be used in the production. Secondly, the number of parameters of a model varies with the increase the model's complexity. Thirdly, these systems still require a lot of manual interactions.

Dym et al. (1988) and Han (1997) addressed the necessary of drawing interpretation in the construction industry. Specifically, they wrote "a computer system which could automatically recognize the individual elements by interpreting patterns of lines, arcs and strings would be an interesting and challenging task." However, research in the area of engineering drawing in the construction industry is a new subject. Therefore, pertinent literature is limited. Only a few systems for the recognition and interpretation of constructional engineering drawings have been reported (Ah-Soon and Tombre. 1997) in which most of them are concentrated on recognition and understanding of architectural engineering drawings (Koutamanis e al. 1989, Ryall and Shieber, 1995, Aoki, Shio, 1996, Ah-Soon 1997, Huang and Pridomore(1997). Koutamanis (1989, 1992, 1993) is perhaps the first researcher focusing on drawing understanding and interpretation in the construction industry. Other examples include the OGAR (Querying Graphics through Analysis and Recognition) team of LORIA at France who developed an engineering interpretation model named **CELESSTIN** and applied it in architecture drawings to reconstruct 3D models of a building (http://www.loria.fr/equipes/isa/index_anglais.html). A team at the University of Carnegie developed a system called SEED (Software Environment to Support Early Phases in Building Design). This system is to provide support, in principle, for the preliminary design of buildings. Drawing interpretation technologies are also applied in a system for room searching and spatial analysis (Flemming, Woodbury, and Coyne 1993). Lewis and Sequin at the University of

California at Berkeley have developed the Building Model Generator (**BMG**) to generate 3D building models from 2D architectural plans for a smoke-spread prediction model, or CFAST, which is developed by NIST (Lewis and Sequin 1998).

Huang (1997) firstly presented the concept for using drawing interpretation for automatic taking off, and an experimental system was developed. However, the system is also based on approximate models of architectural plans, thus its results are only useful for preliminary cost estimation.

In order to reduce the time and to improve the efficiency in steel reinforcement measurement, a knowledge-based system is developed to recognize drawing elements from structural drawings, and then measure the quantities of steel reinforcement automatically. Specifically, this chapter describes the rule-based graphics recognition approaches used in developing the system, together with illustrations of how structural reinforcement elements are automatically recognized and measured. The system consists of three modules: VHFraming, VHDetails, and VHQuantity. VHFraming is designed to recognize the framing plan of a structure. VHDetails is designed to recognize the steel reinforcement in the drawing of a particular structural elements such s column, beam, wall, slab, staircase, etc. Finally, VHQuantity combines the results of the two programs and generates the bending schedule of steel reinforcement. VHStation adopts the technologies on graphic recognition and rule-based approaches. It assimilates quantity surveyor's professional knowledge and experience into rules to formulate rule-based systems, which significantly improves the efficiency and accuracy of the quantity surveying process.

7.2 Knowledge Representation in VHStation

For drawing understanding and interpretation a-priori knowledge must be embedded into the interpretation systems. Interpretation can be taken as the process of applying knowledge to given data in order to obtain a meaningful description of the data. Domain-dependant knowledge can be separated into disjunctive parts. Main kinds of knowledge are: knowledge about objects and their properties, knowledge about structural relations and knowledge about geometrical constraints.

According to the knowledge applied to them, the process of understanding and interpretation of drawings can be divided into three phases: lexical phase, syntactic phase and semantic phase (Tombre 1997). Lexical phase is mainly the primitive recognition phase. In the syntactic phase rules and standards are used to check the correctness of a drawing. In the semantic phase knowledge about structural relations are used to combine objects understood in the first two phases to high-level objects. This phase also checks whether a drawing represents a feasible object or not.

Knowledge is the basis of recognition. How to apply knowledge for recognition can be broadly categorized into following classes, based on the type of knowledge description:

(1) Algorithm: The knowledge of recognition is embedded into algorithms of the system (Lwata et al. 1988, Suzuki et al. 1990, Boatto et al. 1992, Ablameyko et al. 1993, Ogier et al. 1993).

(2) Structural description: The recognition system defines the recognition procedure in several modules. The system controls the sequence and relationships between different modules. But the kernel model about reasoning is still described in algorithm (Joseph 1992, Pasternak 1995).

(3) Syntactic/Semantic description: Knowledge is described in standard language or data structure. An independent algorithm performs the recognition task by searching for targets that match the description. Some reasoning is also required in the searching procedure (Dori 1989, Pasternak 1993).

In VHStation all the knowledge of structural drawing interpretation is represented as rules and saved in the rule database. Rules were obtained from two approaches: Manually crafted, and automatic acquisition with a rule generation mechanism.

7.3 Characteristic of Structural Drawings

Structural engineering drawings in construction can be divided into two classes: framing drawings and details drawings. Framing drawings provide the top view of each floor of a building while details drawings show the structural detail of each construction element such as beams, columns, slabs, walls, staircases and water tanks. In many aspects, structural drawings are similar to mechanical engineering drawings, as they typically represent orthogonal projections of the solid elements. But there are also many differences. The first is that some information in structural drawings may not be explicitly indicated in the drawings, but can be interpreted according to the relative rules and engineer's experience. For example, engineers usually draw only one polyline of steel reinforcement bar to represent a group of them having the same attributes but different locations. Secondly, structural representations in the structural drawings are very heterogeneous. For example, the drawing of a wall can be given by its top view, or its front view, or its side view, or using only textual descriptions. Thirdly, some typical drawings are usually provided and referred by many elements having the same structures but different sizes. In addition, frequently there are errors in framing and details drawings;

and these errors can only be found and corrected by combining information from both framing and details drawings.

7.4 Design Considerations

In order to make VHStation popular in the Q.S. field, it is necessary to include correct development strategies, appropriate functions and friendly user interface in the system design.

7.4.1 Development Strategies

The system adopts its development strategies in the following three aspects: data source, data content, and application concepts:

a. Data Source

Although nowadays quantity surveyors perform their tasks using paper-based drawings, VHStation uses electronic drawings as input. Since drawings are prepared by some kind of CAD software such as AutoCAD, it is important for VHStation to adopt the commonly used data format particularly the DXF and DWG formats. This does not only save a lot of time in plotting and copying mass amount of drawings but also guarantees data integrity. As for those construction projects that do not have electronic drawings, a Vectorization software can help to convert paper drawings into DXF/DWG format file before applying VHStation for the taking off process.

b. Data Content

Most of the source data for the taking off comes from both framing plan and details drawings. Framing plan drawings show the overall layout of a building usually in a floor-by-floor basis, while details drawings show the detailed steel reinforcements design of each structural element. As such, VHStation chooses framing plan drawings and details drawings as the two object sources to be processed. There are also small amounts of information scattered around in different drawings such as the typical drawings or some general notes within the detail drawings that are essential in the calculation but relatively difficult to be located automatically. The user usually inputs this information manually, thus system process efficiency and information integrity is guaranteed.

c. Application Concepts

The application concept is to design a system as close to a human Quantity Surveyor as possible and try to integrate the current practice into the system. All the calculation and method of measurement is based on British Standard which is commonly adopted in Hong Kong. Also, the drawing format may be different for different designers, the wording may varies from draftsman to draftsman. In order to make the software widely acceptable by the industry, efforts have been made to incorporate these variations as much as possible. This is an endless task, as the drawing format has not been standardized. Nevertheless, the present system possesses great deal amount of structural engineers and quantity surveyors expert knowledge and is able to produce very accurate results of bending schedule with current used format.

7.4.2 Appropriate System Functions

Based on the characteristics of structural engineering drawings being processed, the following points are addressed in the design of system functions:

a. Due to the fact that the steel reinforcement information is presented by the combination of geometric figures and descriptive text in the drawings, there are certain physical connections among the text and the drawing. Therefore, in order to extract the steel reinforcement information wholly, the system must bear the functions of powerful graphics recognition, high level of text understanding, and interrelation ability.

b. There are a variety of ways in representing basic graphical primitives in structural drawings and various methods of combining those basic primitives into meaningful knowledge. Consequently, analyzing techniques based upon artificial intelligence must be introduced to analyze and recognize the drawing elements in different hierarchy, particularly in their relationship, as to understand the drawings.

c. In most cases, the structure of steel reinforcement is represented in two different projection views, namely the front view and the section view. For example, a R.C. beam is usually shown by a front view with one or more section views with all the reinforcement bars represented by a line or dot respectively. Hence the system must have the certain ability of 3D-reconstruction in order to recognize the spatial shape and dimension information of each of the steel reinforcements being used.

d. The system must also be able to affirm the reliability level of the calculated results and provide hierarchical post-processing functions because the number of steel reinforcements is so large that it is too difficult to check them one by one. This reliability check is displayed in various means to the user to express the level of accuracy of the data so that in case of uncertainty, the user may choose to double check the data manually.

e. Along with the progress of a project, some drawings may have version updating. Accordingly, the system must provide the function of version management, comparing the difference of drawings of different versions and find the difference of quantity surveying results. This is particularly useful in the construction industry since variation of design is unavailable in large-scale project and quick updating may save numerous time and money.

f. The system must be able to find and display abnormal data to help users to identify and correct them because most drawings contain certain errors. A human can easily deal with these errors but for a software system, this must be identified and confirmed by the user in a very user-friendly manner so as to increase system efficiency.

7.4.3 Friendly User Interface

The friendliness of user interface is mostly about the easiness of learning and using the system. The correctness and efficiency of the system is particularly important and depending on the user interface. In order to provide a friendly user interface, VHStation makes great effort in the following aspects:

a. The operation mode provided is close to the language use of a quantity surveyor. The interaction ought to be intuitionist, convenient, and an instant observation of the result is provided for easy checking.

b. The system displays the calculated result in tabular form, which is clear at a glance. The correspondence between each item in the table and the primitives in drawing can be consulted and revised easily.

c. Due to the complex nature of the structural drawings, the following three cases on drawing recognition may arise: (1) None standard drawing manner that is unknown to the program. (2) Data distribution exceeds the processing range. (3) Incompleteness of data due to the drafting error. These will either require the user to check and affirm the result or the user will be unable to obtain a meaningful result. Once these situations are detected, the system will highlight them and let the users add command inputs to the system. This will minimize the users' effort in checking and verification.

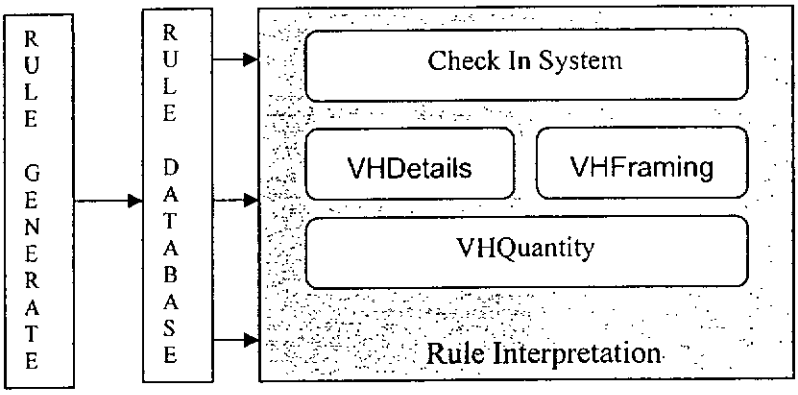


Fig. 7.1 Architecture of VHSTATION

7.5 The Overall Architecture of VHSTATION

The overall architecture of the VHSTATION is illustrated in Figure 7.1. The task of drawing interpretation starts after paper-based drawings are scanned and vectorised. During the pre-processing phase, a series of operations are applied to the vectorised drawings to analyze the graphical primitives and the layout of the drawings. The interpretation phase is the core of the system. During this phase, string, symbol, steel reinforcement line and other elements are interpreted based on a set of generic rules.

In VHSTATION feature-based generic rules are used to represent the knowledge for interpretation. The feature based generic rules enable the system to have more flexibility in handling the variations of drawing elements, and the rules can be easily modified to match new conditions. This chapter focuses on the interpretation phase of VHSTATION to demonstrate how to use feature based generic rules to analyze framing and details drawings. The detail algorithm of Check In System and Rule generation could be found in chapter 3 and chapter 6.

7.6 Interpretation of Drawings Using Feature-Based Rules

Recognition and interpretation of engineering drawings requires identifying of graphical features of drawings. Tikerpuu and Ullman (Tikerpuu and Ullman 1988) define that features of graphics are geometric shapes that can be expressed in general parameters and can be ratiocinated in applications. Cunnigham and Dixon (Cunnigham and Dixon 1999) point out that features can be described using three components: feature syntax, feature semantics, relationship between syntax and semantics.

In developing VHSTATION, the features in construction engineering drawings are classified into the following levels.

1. Pixel level: the most primitive representation

2. Graphics primitives level: the representation of graphics primitives such as line, arc circle and so on
3. Symbol level: representation of a group of graphics primitives
4. Element level: representation of a structure element
5. Frame level: Representation of the whole structure of a building

In this chapter the research focuses on the symbol and element levels, as the other levels of features are either irrelevant, or can be further divided into symbol and element levels.

7.6.1 Framing Drawing Interpretation

To calculate the quantity of steel reinforcement of a building, the framing and details drawings of a building must be analyzed and understood together. Normally, the

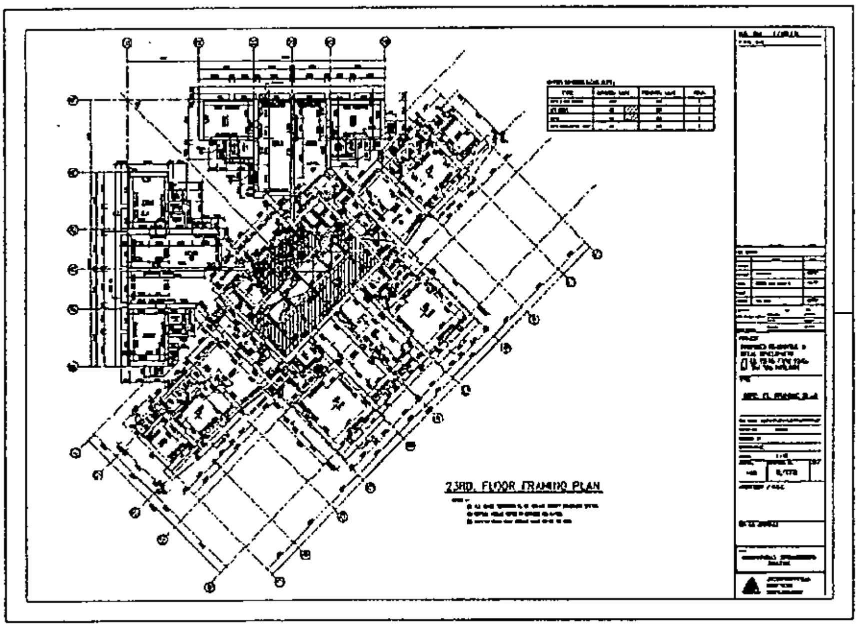


Fig. 7.2 Framing Drawing

topological information of the construction elements can be extracted from the framing drawings while the quantity of steel reinforcement of each element can be calculated from relevant details drawings. A typical framing drawing is shown in Fig. 7.2

In Fig. 7.2, the construction elements are expressed directly by graphical primitives. The name of an element is marked as a text string and the element shape is represented by its boundary, which may be a rectangle, a circle, or a close-chain of lines and arcs. Although the text strings and associated graphic elements are normally explicitly indicated, sometimes this part of information may be omitted, thus making the process of automatically recognizing drawing elements a more difficult task. The interpretation procedure of Framing Drawing is shown in Fig. 7.3.

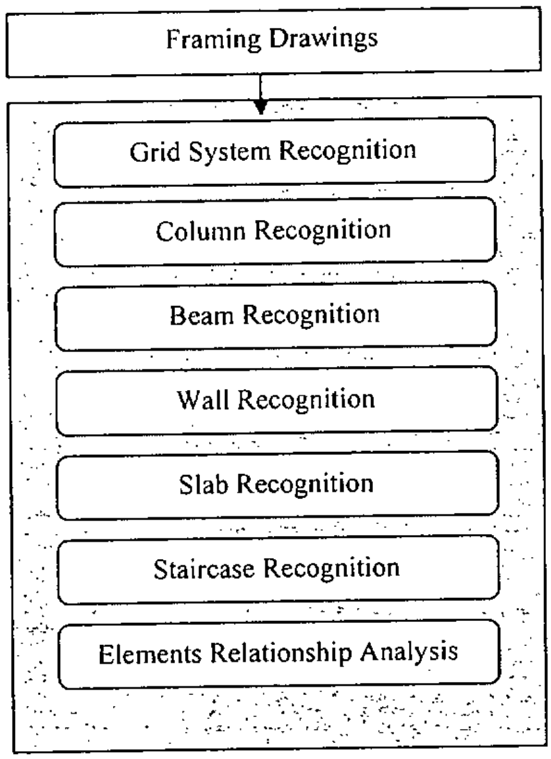


Fig. 7.3 Framing Drawing Interpretation Procedure

- Recognition of Coordinate System of the Framing Drawing

Each framing plan drawing has one or more coordinate systems, which are used throughout the project. Through them, the distribution and dimension of each structural element are described and identified. Coordinate systems are the basis of recognition of framing plan drawings.

Coordinate system is generally an orthogonal system, and occasionally a polar system. Sometimes both of them appear in the same framing drawing. VHFraming obtains coordinate lines through two groups of coordinate symbols, {'A', 'B', 'C', } and {'1 , '2 , '3 , }. These symbols are embedded in circles and each circle is connecting one dot-dashed line. They make grid-lines and establish the coordinate system.

- Sequence of Element Recognition

Primary types of structural elements in framing drawings are BEAM, WALL, SLAB, STAIRCASE and COLUMN. Each element is shown in an outline format in a framing plan drawing. The visible part of the outline is drawn in solid lines while the invisible part is drawn in dashed lines. Each element outline has certain relation with coordinate lines. For example, the center of a column is commonly located at the grid point of the coordinate system. Beams include two classes: "beam on column" and "beam on beam"; the former is normally located between two adjacent grid points while the later is normally parallel to a grid line. Outlines of a slab lie on beams or walls, etc. Based upon those geometric relations, VHFraming recognizes elements in a sequence of "column - beam - wall - slab - "

- Methods of Element Recognition

Considering the characteristics of graphical expressions, difficulties in recognizing with framing drawings can be summarized as follows:

- Farraginous Geometric Elements: Quite many graphical primitives that are useless for recognition mix with those useful ones and outlines of different elements are also mixed up.
- Individual Vagueness: Outline of a single element always has no strict geometric representation, but only needs to have correct shape visually.
- Group Vagueness: Outlines of different elements and also different types of elements interweave with each other. Their dubieties are full of high-level construction semantic implications that are very important for recognition.

In other words, the difficulty lies in how to extract precise and integrated information of the construction element from those vague, imprecise and incomplete geometric information (geometric element and the relations among themselves) under the support of domain knowledge in construction quantity survey field.

- Recognition of Each Element

Each element is represented by its outline, its characteristics of position and name. The name is expressed as a string. It denotes the semantic meaning of its related outline which is expressed as some kinds of basic geometric elements such as lines and indicates the geometric characteristics of the element. This shows that understanding the relation among these aspects is important. Consequently, the recognition begins with recognizing the above primitives and then makes decisions after synthesizing them.

For instance, the condition of determining a column can be expressed as below:

Rule 1: A column expressed in framing drawings contains three parts: Column Mark, the relativity between Column Mark and Column Boundary.

Rule 2: A Column Mark is a string, its length is no more than 5 and each character of the Column Mark belongs to the set of normal 26 christcross-row or number.

Rule 3: The relativity between Column Mark and Column Boundary is the distance between them. The distance should be within a possible range.

Rule 4: The Column Boundary has three elements: the position of the boundary, the size of the boundary and the shape of the boundary.

Rule 5: The position of the column should be at the Grid-Intersection or any other special position.

Rule 6: The shape of column can be rectangle, circle, ellipse or any head-tail connected lines or arcs.

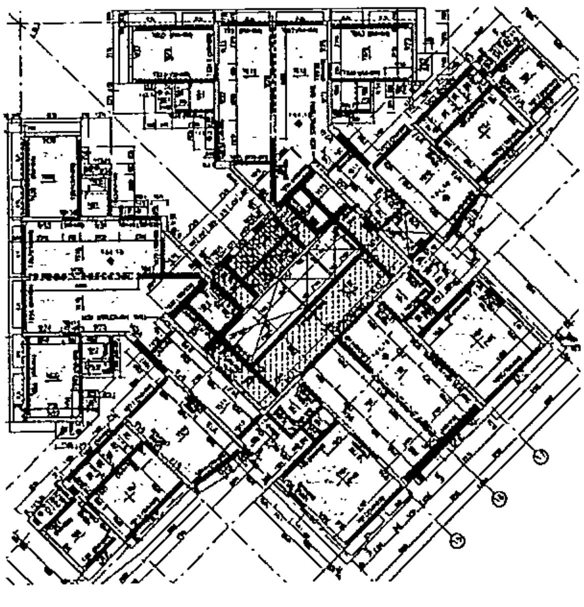


Fig. 7.4 Framing Drawing Recognition Result

Rule 7: The Grid Intersection can be defined as the intersection of the Grid Line.

Rule 8: The Grid Line is the line that connects the Grid Circle.

Rule 9: The Grid Circle is the circle that has a Grid Mark inside.

According to the above rules, recognizing a column is started from the extraction of graphical primitives, which can be a rectangle, circle, ellipse etc. If some primitives match the relevant rules then these primitives form a set. If the set matches the definition of column, then a column is recognized. An example of the recognition result is shown in Fig. 7.4

7.6.2 Details Drawing Interpretation

It shows all the details that are needed to construct the part. Each part in the framing drawing will have a corresponding detail drawing with all the information necessary for its construct. Therefore, a detail drawing must describe the shape and the size of the part, as well as provide all the specifications that are necessary for its production. For example, a typical beam details drawing of a beam is shown in Fig.7.5.

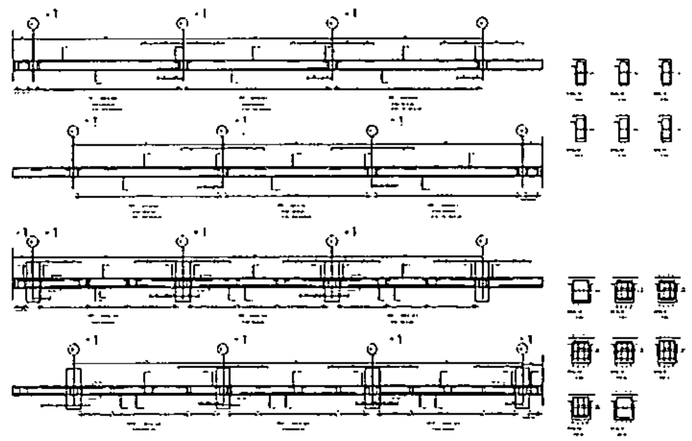


Fig.7.5 Example of Beam Details Drawing

In order to automatically recognize and interpret the information contained in the details drawings, VHSTATION has to recognize two types of information: architectural symbols, and reinforcement steel. Techniques used in recognizing these two types of information are described as follows. And the interpretation procedure is shown in Fig. 7.6.

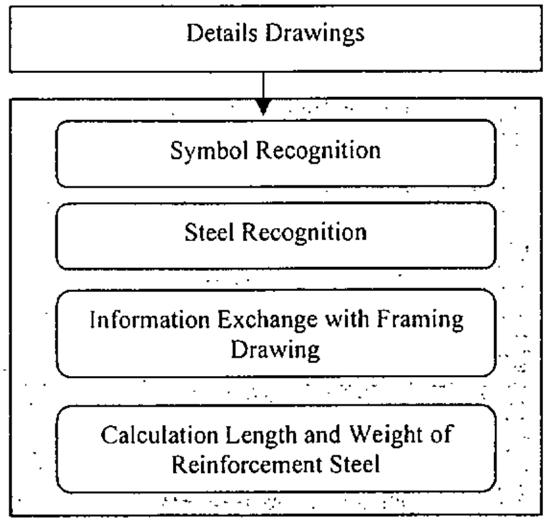


Fig. 7.6 Interpretation Procedure of Detail Drawing

7.6.3 Symbol Recognition

There are many kinds of symbols used in details drawings such as model matching, decision tree, neural network, graph, matching, syntactic/structural matching and statistical method (Lladós and Martí. 1999, Ah-Soon, Tombre 1998, Okazaki et al. 1988). And the recognition of symbols is conducted on three levels: image level, graphical level, and syntactical level (Tikerpuu and Ullman 1988). Syntactic/structural approach is the most widely used approach in symbol recognition. However, syntactic/structural approach cannot represent certain geometric properties, such as the size of the primitive, relative orientation, edge concavity, and the like, that are important for distinguishing among features (Qiang et al., 1997). These limitations will prevent it to be used widely in the field of structural drawings interpretations.

In this chapter, symbol recognition is only discussed on the syntactical level. This is because the unique characteristics of symbols used in civil engineering design. The major difference between symbols used in civil engineering (engineering symbols) and other symbols is that engineering symbols have multiple forms. This means that one symbol can have several different outlooks. Therefore, interpretation techniques based on the image or graphical level will not be able to accommodate the task of recognizing engineering symbols. The syntactical approach, however, can solve this problem.

Generally there are five types of symbols presented in details drawings: section symbols, height symbols, truncation symbols, begin-end symbols, and middle symbols. The recognition of a section symbol is used as an example to explain the mechanism used in recognizing symbols by the syntactical approach. Figure 7.7 shows some typical section symbols used in structural drawings.

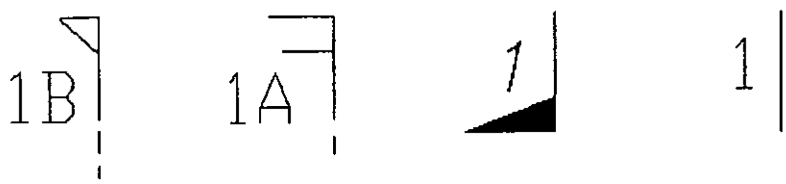


Fig.7.7 Section Symbol

The rules for interpreting section symbols are described as the following:

Rule 10: A standard section symbol contains five elements: Cut Direction Line, Auxiliary Cut Direction Line, View Direction Line, Auxiliary View Direction Line and Section Text. The length of the line in section symbol is less than one text height.

Rule 11: Cut Direction Line is the line whose length is less than two text heights and greater than one text height, and the line is at the cut direction.

Rule 12: Auxiliary Cut Direction Line is the line that is in parallel with the Cut Direction Line and its length is less than one text height.

Rule 13: View Direction Line is the line whose function is to show the viewdirection. It is perpendicular to the Cut Direction Line and its length is not greater than the length of the Cut Direction Line.

Rule 14: Auxiliary View Direction Line is the line whose function is to show the direction of the View Direction. Its length is less than the length of ViewDirection Line.

Rule 15: Section Text is a string. Each character of the string is in the set of normal 26 christcross-row or number.

Figure 7.8 to Figure 7.11 illustrate how a section symbol is extracted.

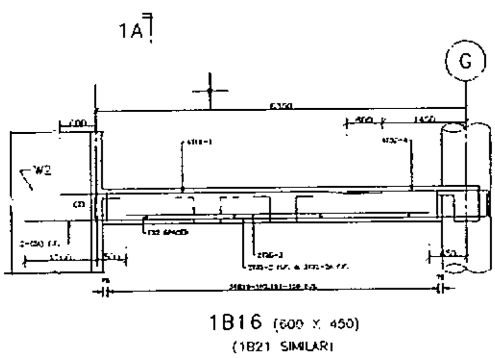


Fig. 7.8 Original Drawing

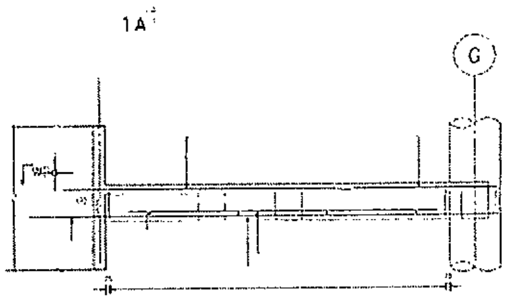


Fig. 7.9 Drawing processed after Rule 15

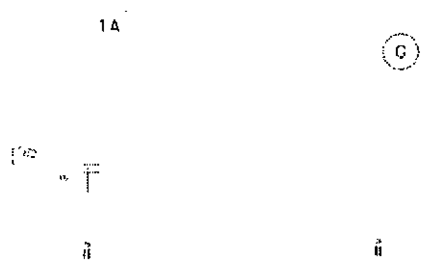


Fig. 7.10 Drawing processed after Rule 10 **Fig. 7.11 Drawing processed after Rule 11 to14**

The procedure of extracting a section symbol can be described as follows,

- Firstly, VHSTATION removes the strings that are not section text based on Rule 15.

• Secondly, VHSTATION searches for the arc near the left string and removes the long lines that could not be the lines of section symbol, based on Rule 10.

• Thirdly, VHSTATION checks the lines that near the left string and decides whether the lines contain Cut Direction Line and View Direction Line based on Rule 11 to 14.

As the result of completing the above procedure, the section symbol is identified and illustrated in Fig. 7.8.

7.6.4 Reinforcement Steel Recognition

Rules are also used in interpreting reinforcement steel. The reinforcement steel usually has the following three components: annotation string, polyline and connection line. An annotation string indicates the steel type, diameter, amount, serial number, and location of reinforcement steel. A polyline represents the shape of steel reinforcement. The connection line is used to connect the annotation string with the polyline. For example, the annotation string "5Y10-200 T&B" represents five steel bars, which are located at every 200-millimeter gap between each other at both the top and bottom of the construction element. Rule 16 is used to express the main elements of reinforcement steel.

Rule 16: Reinforcement steel contains five elements: Steel-Annotation-String, Mark-Line-Relativity, Annotation-Line, Line-Bar-Relativity and Steel Bar.

(Note: 'mark-line-relativity' is used to measure the closeness between an annotation string and an annotation line, and the 'line-bar-relativity' is used to measure the closeness between an annotation line and a steel reinforcement bar. Although the format of an 'annotation-string' is well defined, there exist a number of variations that are acceptable to the profession.)

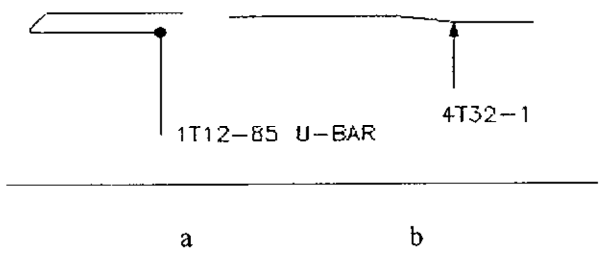


Fig. 7.12 Reinforcement Steel

Fig. 7.12 gives two examples of reinforcement steel: "1T12-85 U-Bar" and "4T32-1 . The line that is near "1T12-85 U-Bar" and is linked with a dot is an Annotation-Line. And the polyline intersected with the Annotation-Line at the center of dot is the Steel Bar.

In order to automatically interpret the information of reinforcement steel, the system needs to firstly extract the Steel-Annotation-String from the details drawing. The interpretation rules are shown as the following.

Rule 17: A Steel-Annotation-String has five main elements: ST-Amount, ST-Type, ST-Diameter, ST-Number, ST-Location-Attribute and Other-Notation.

Rule 18: ST-Amount indicates the amount of reinforcement steel.

Rule 29: ST-Type shows the type of reinforcement steel. It can be 'T' or 'Y' or 'R' or 'ET'.

Rule 20: ST-Diameter gives the diameter of reinforcement steel bars. ST-Diameter should be an integer. The value ranges from 10 to 40.

Rule 21: ST-Number is the serial number of reinforcement steel. ST-Number should be an integer or integer plus a character.

Rule 22: ST-Location-Attribute: ST-Location-Attribute gives the location of reinforcement steel. It can be a sting or sentence. There are several typical ST-Location-Attributes like "T1& B1 , "E.F", "T2 , "B2 . T means top of the slab and B means bottom of the slab. "E.F" means each side of the elements.

Rule 23: Other-Notation can be a symbol or a sentence. The function of Other-Notation is to describe extra information of reinforcement steel.

The procedure for interpreting the Steel-Annotation-String is shown in Fig. 7.13.

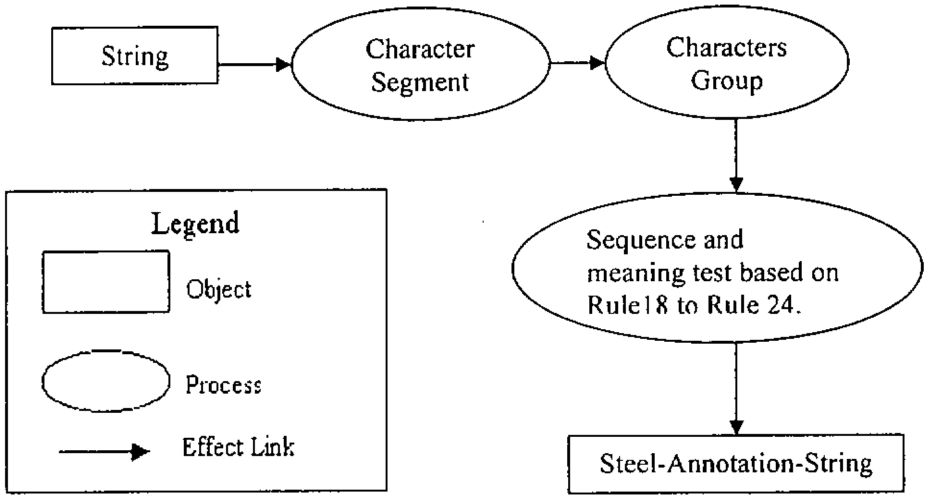


Fig. 7.13 Subsystem for Understanding Steel-Annotation-String

Let us use examples to explain how the interpretation process is conducted. Firstly, for interpreting Steel-Annotation-String, the string is separated into individual symbols. For example, "5Y10-200 T&B" can be separated into "5,Y,1,0,-,2,0,0, ,T,&,B". This string of symbols are checked and interpreted based on Rule 18 to Rule 24.

Determining the correct relationship (Mark-Line-Relativity) between the Steel-Annotation-String and the Annotation-Line is an important yet difficult task. Because the complexity of engineering drawings, the Mark-Line-Relativity is very difficult to identify. There are normally 4~8 similar lines around the Steel-Annotation-String, among which one is the Annotation-Line. The Nearest-Distance method is the traditional method used to judge the relationship between a line and a string. But this method only considers the distance between line and string, and does not take account of the impact of symbols on the line and the collocation of string and line.

In this study a novel method has been developed to solve the problem, called enhanced gravity field relationship method (EGFRM). In the method each line is assumed to have a gravity field. The relationship between the Steel-Annotation-String and Annotation-Line can then be simplified as the nearest gravity relationship. However, the gravity field of line in VHSTATION is not exactly the same as a normal gravity field, as under most circumstances a normal gravity field cannot represent the correct relationship between drawing objects. So two additional factors are used: symbols on the line and collocation of string and line, to modify the gravity field of a line. Symbols that affect the gravity field of a line include short line, dot and arrow on the line. These two factors can change the size, shape area and orientation of the gravity field. Some sample points are selected in a gravity field to shape the gravity of a line. The positions of the sample points are shown in Fig. 7.14. The number and positions of sample points can be changed to suit different conditions.

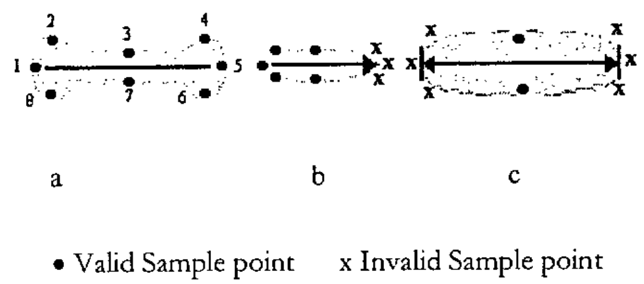


Fig. 7.14 Gravity Field Impacted by Symbol

Fig.7.14 gives examples of the gravity field on a line and how arrows and dimension symbols change the gravity filed of a line. Fig. 7.14(a) is the normal gravity field of a line. All the 8 points are taken into account in determining the relationship between a line and a string. If the line has an arrow at one end, then the text that connects this line is usually near the other end. So the gravity field of the arrow end is minimized. As Fig. 7.14(b) shows that points 4, 5,6 are not used in relationship decision. Based on the same reason, if the line has dimensional symbols at the two ends, then only points 3, 7 are used in relationship decision.

Collocation of a string and a line also changes the gravity field of a line. Fig. 7.15 gives an example of how the collocation changes the gravity field of line. In Fig. 7.15 three lines and Steel-Annotation-String are arranged in layers. If using the normal gravity field of a line, it is difficult to decide which line is connected with the string "3Y10-91-300 B2 .

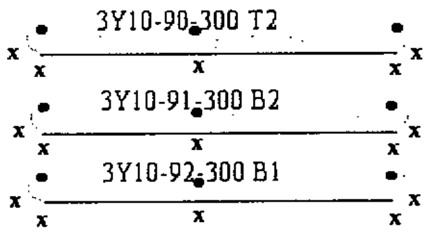


Fig. 7.15 Gravity Field Impacted by Collocation of Strings and Lines

In VHSTATION, if a group of lines and texts are grouped, then the gravity fields of lines in the group will be changed according to the arrangement. For example if the arrangement is that lines are on the top of each other as shown in Fig. 7.14, then only sample points 2, 3, 4 will be used in deciding the relationship between lines and strings.

After forming the enhanced gravity field of a line, Mark-Line-Relativity can be identified between the Steel-Annotation-String and Annotation-Line. Before identifying the relationship function D(P,Q) is defined to represent the distance between two points P and Q.

The procedure of identifying the Mark-Line-Relativity can be described as follows,

- Firstly, VHSTATION recognizes the enclosed rectangle of Steel-Annotation-String. Then VHSTATION obtains six points from the enclosed rectangle. The six points include the four corner points and two middle points of the long edges of the rectangle.
- Secondly, VHSTATION calculates the distance between the Steel-Annotation-String and each Annotation-Line by using function $D(P_i,Q_j)$ ($1\leq i \leq 6$ & $1\leq j \leq 8$). P_i is the set of points on the enclosed rectangle of the Steel-Annotation String. And Q_j represents the set of points in the gravity field of the Annotation-Line.
- Lastly, select the lowest value of the function, then the corresponding Annotation-Line can be regarded as associating with the Steel-Annotation-String.

After selecting the correct relationship between the Steel-Annotation-String and the Annotation-Line, the next task is to identify Steel Bars. There are two types of Steel Bars: Intersectional Steel Bars and arrow-point Steel Bars. Fig. 7.12 (1) is an example of the Intersectional Steel Bar and Fig. 7.12 (2) is an example of the arrow-point Steel Bar. Rules 24 to 26 explain how VHSTATION identify Steel Bars.

Rule 24: Steel Bar has two types: Intersectional Steel Bar and arrow-point Steel Bar.

Rule 25: Intersectional Steel Bar is a polyline intersected with Annotation-Line, and the intersection point is marked by a dot symbol.

Rule 26: Arrow-Point Steel Bar is a polyline intersected with Annotation-Line, and the intersection point is pointed by an arrow.

Rule 27: Steel Bar and Annotation-Line must be perpendicular at the intersection point.

Rule 28: If the Steel Bar is an Arrow-Point Steel Bar, then the distance between intersection point and arrow head is less than the distance between intersection point and arrow tail.

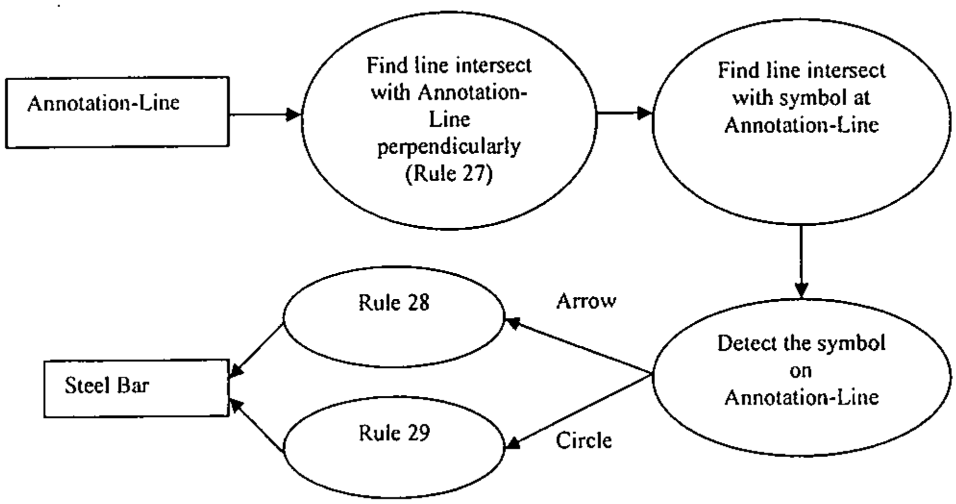


Fig. 7.16 Subsystem for Detecting Steel Bars

Rule 29: If the Steel Bar is Intersectional Steel, then the distance between the intersection point and center of the circle is less than half of the diameter of the circle.

The procedure for interpreting the Steel Bar can also be shown in Fig. 7.16. After recognition of steel reinforcement, the structure is of each steel reinforcement is analyzed based on the BS4466 standard and attached with a shape code. The length calculation of the reinforcement is based on the attached shape code.

7.6.5 Rules for combining Framing Drawings and Details Drawings

After interpreting framing drawings and details drawings, the most important work is to combine the information together to get the correct dimensions and quantities of reinforcement steel. Because there may be various discrepancies and errors of dimensions in construction drawings, the interpreted dimensions cannot be used directly. VHSTATION uses a verification mechanism to compare and produce correct dimensions. These correct dimensions are subsequently used to calculate the quantities of reinforcement steel. The rules used for verification and calculation are given below:

Rule 30: If dimensions of a drawing element in framing and details drawings are the same, then either dimension can be used in calculation; else refer to Rule 25 to Rule 27.

Rule 31: Gird dimensions should be derived from framing drawings.

Rule 32: Top view dimensions should be derived from framing drawings.

Rule 33: Front view or section view dimensions should be derived from details drawings.

Rule 34. If there exists a string to describe the size or length of an element, then corresponding dimensions should be derived from the string.

Rule 35: Correct dimensions of reinforcement steel can be derived from the combination of following: dimensions in framing drawings, dimensions in details drawings and relevant formula.

Rule 36: The height of an element can be derived from its height symbols.

Rule 37: Combination methods include: +, -.

Rule 38: Combination is based on the Grid line of the drawing and the name of each element.

Rule 39: All the dimensions in framing drawings and details drawings are combined together to form a 3D dimension grid. If the line of reinforcement steel can be projected to a dimension, then the length of the line is the value of the dimension.

Rule 40: The concrete cover should be considered in the calculation.

Rule 41: If the length of reinforcement steel is less than the minimum requirement, then the minimum requirement should be used.

Rule 42: If the length of reinforcement steel cannot be obtained from above rules, it is calculated on scale.

The above rules enable the system to combine the information recognized in framing and details drawings in order to derive the dimensions and quantities of reinforcement steel.

7.7 Conclusions

In this chapter, the following issues are addressed in the development of VHSTATION:

- Due to the fact that the steel reinforcement information is presented by combination of geometric symbols and descriptive texts in drawings, there are certain physical connections among texts and symbols. Therefore, in order to extract steel reinforcement information, the system must be able to recognize symbols, as well as related texts.

- There are a variety of ways in representing basic graphical primitives in structural drawings and various methods for combining those basic primitives to represent engineering elements. Consequently, analyzing techniques based upon artificial intelligence must be introduced to analyze and recognize the drawing elements in different hierarchy, particularly in their relationship, in order to understand the drawings.
- In most cases, the structure of steel reinforcement is represented in two different views of projection, namely the front view and the section view. For example, a R.C. beam is usually shown by a front view with one or more section views with all the reinforcement bars represented by a line or dot respectively. Hence the system must have the ability of partial 3D-reconstruction in order to recognize the spatial shape and dimension information of each of the steel reinforcements being used.
- The System must also be able to affirm the reliability level of the calculated results and provide hierarchical post-processing functions because the number of steel reinforcements is so large that it is too difficult to check them one by one. This reliability check is displayed in various means to the user to express the level of accuracy of the data so that in case of uncertainty, the user may choose to double check the data manually.
- Along with the progress of a project, some drawings may have version updating. Accordingly, the system must provide the function of version management, comparing the difference of drawings of different versions and finding the difference of quantity surveying results. This is particularly useful in the

construction industry since variation of design is unavoidable in large-scale projects and quick updating may save enormous time and cost.

- The system must be able to find and display abnormal data to help users to identify and correct them because most drawings contain certain errors. A human user can easily deal with these errors but for a software system, this must be identified and confirmed by the user in a very user-friendly manner so as to increase system efficiency.

Existing drawing interpretation systems mainly tackle single type of drawings (Ah-Soon et al. 1997; Joseph 1992; Larry 1999; Fahn 1988). This study presents a rule-based drawing interpretation system which can process different types of drawings: framing and details drawings which have differences in cartography and display formats. The rules used in interpretation are based on features identified in the construction engineering drawings.

The implementation of VHSTATION is completed, and the system is commercially available. Interesting parties may contact the author for further information about this system.

Chapter 8

MINIMIZATION OF DESIGN ERRORS USING 3D RECONSTRUCTION AND COLLISION DETECTION TECHNIQUES

8.1 Introduction

Rework is the unnecessary effort of re-doing a process or activity. It occurs when a process or product does not meet the requirements of the customer (Love, Li and Mandal, 1998). Consequently the product is required to be altered in accordance with customers' requirements. Research (Li and Love 1998, Love, Li and Mandal 1998) has found that the additional cost to construction caused by rework can be as high as 12.4% of total project costs in the Hong Kong construction industry. Among the major causes of rework, design errors are identified as a major source (cause) of rework in construction, contributing to 51% of the total failure costs of rework (Li and Love 1998, Hammarlund and Josephson 1991). Among the design errors, most of them are attributable to dimensional inconsistencies presented in different sets or sections of engineering drawings (Mondal et al. 2000).

2D representation of engineering drawings are unintuitive, difficult to manipulate and modify, and do not facilitate identifying dimensional errors (Liu, et al. 2001). It is therefore important to posses the capability in reconstructing 3D models from 2D drawings. This chapter presents a hybrid use of 3D reconstruction and collision detection techniques to assist designers and construction professionals in detecting dimensional inconsistencies in drawings before the execution of construction activities.

Over the past two decades, many algorithms for reconstructing 3D models from orthographic 2D views have been developed. These algorithms can be classified into two major approaches: the CSG (Constructive Solid Geometry) oriented approach and the B-rep approach. The CSG approach adopts a top-down reconstruction strategy which assumes that each 3D object can be built from certain primitives in a hierarchical manner. It starts with selecting a 2D loop as a base and uses translational sweep operations to construct 3D primitives; the constructed primitives are then assembled to form the final object (Meeran and Pratt 1993, Shum et al. 1997). A major drawback of the CSG approach is its inability to recognize special patterns in the primitives generated from complicated drawings. Thus this approach is only suitable for constructing parts of uniform thickness or rotational objects (Liu et al. 2001). The B-rep approach bases itself on the concept of 'fleshing out projections (Wesley and Markowsky 1980). This approach firstly generates 3D candidate vertices, 3D edges and 3D faces from 2D vertices and edges. The 3D objects are then constructed from combining candidate faces. Several researchers have implemented variants of this approach (Idesawa 1973, Wesley and Markowsky 1981). Current research efforts have been focusing on extending the range of objects to be constructed.

The task of collision detection can be formally defined as: 'Given two objects and their relative positions, decide whether the objects have interpenetration in the given space". Solutions of this problem will be useful to determine inter-object dimensional inconsistencies in drawings. Many techniques have been developed for collision detection (Kitamura et al. 2000). One class of the techniques is based on the method of performing intersection/interference tests of the objects. This method is simple but it does not work well if objects are only in contact. Another class of techniques is to compute the volume swept out by the objects when they are positioned in the required locations. A collision is declared if these swept volumes intersect. This method is reliable and effective in handling cases in which two objects are just in contact with each other.

For the techniques based on the swept volume test, most of them concentrate on polyhedral objects. Moore and Wilhelms (1988) presented a method for detecting collision among convex polyhedra based on a clipping algorithm and among flexible surfaces modeled by triangles. Hahn (1988) introduces an algorithm suitable for detecting collisions among objects modeled by polygonal surfaces. Other algorithms for detecting collisions include those developed by Bouma and Vanecek (1991) and Lin and Canny (1998). These algorithms form a basis to determine the most suitable solution to the problem of determining dimensional inconsistencies in building designs.

This chapter presents a study on using 3D reconstruction and collision detection techniques to identify design errors in construction engineering drawings. Specifically, Section 2 describes the characteristics of construction drawings and a 3D reconstruction technique which can automatically recognize and reconstruct 3D models from 2D orthographic views to reveal dimensional inconsistencies within design models. Section 3 presents the collision detection technique developed to identify dimensional inconsistencies between design objects. Discussions and conclusions are given in the last section.

8.2 Identifying Design Errors within an Object Using 3D Reconstruction

Before a 3D model of a construction component can be built, it is important that 2D orthographic views are automatically recognized. A construction drawing comprises structural information of building components by the boundary projections of the components in three orthogonal 2D views including the components' top view F(t), front view F(f), and section view F(s). In each of the 2D view, in addition to the boundary projection, information such as special symbols, descriptive texts are also attached. Thus, the task of recognizing a construction drawing must include understanding both graphical and textual descriptions.

In a construction drawing, a construction component can be decomposed into many graphical primitives (GPs). The recognition process needs to identify all the graphical primitives and their relationships in the component (Dori et al. 1990). Specifically, the recognition process in the research starts with recognizing the projection boundaries of the component by removing the interior graphical primitives representing the steel bars or other auxiliary lines, the symbols, and descriptive texts. For example, Fig 8.1 presents the projection boundaries of a beam from its top view, front view and section view which are obtained by removing the GPs in the boundaries.

Frequently, the projection of a component includes parts of other components' projections in order to show their interconnections. This results in unclosed polygons at the joints, as indicated in Fig 8.1. It is therefore necessary to enclose the polygons. For example, after removing the redundant GPs from the projection boundaries in Fig. 8.1, the projection boundaries of the beam are as shown in Fig. 8.2.

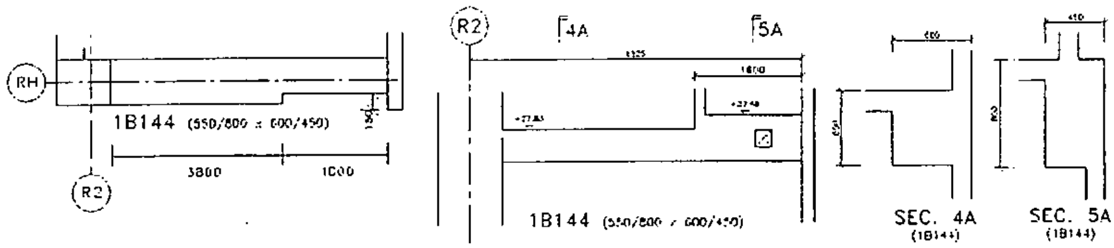


Figure 8.1: Projection Boundaries of a Beam from Top, Front and Section Views

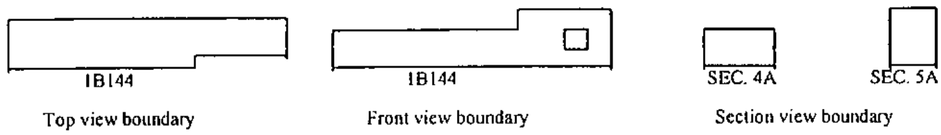


Figure 8.2: Projection Boundaries After Removing Redundant CPs

A construction drawing contains many important symbols and texts. For example, the grid symbols are the reference for view matching; texts describing the sizes of components are indispensable. Thus, after the recognition of projection boundaries the associated information are extracted by analyzing the symbols, texts and boundaries of adjacent components in the drawings.

The extracted data for the projection boundaries, as well as the associated information, are stored in a database structure indicated below.

{

Component name;

Size;

Top view height (if it exists);

Project view;

Group of vertex G (v);

Group of edge G (e);

Group of Grid symbol G (g);

Group of adjacent components G (E);

}

The recognized projection boundaries of the 2D orthographic views can be used to reconstruct a 3D object. Among all the reconstruction methods, the method for reconstructing stretch objects, which belongs to the category of CSD approach, is considered the most suitable and selected, because almost all the building components/objects are designed as stretch objects or a composition of several stretchable objects. A stretchable object is defined as follows. Given the three orthographic views of an object, if there are at least two of the three views have rectangular projection boundaries, then the object is a stretch object, or 2.5D object. A stretchable object can be created by stretching a plane along an appointed track. The plane to be stretched is called the stretch plane (Nezis and Vosniakos 1997).

The stretch method requires breaking down a composite object into a group of simple stretchable objects, or sub-objects, that can be reconstructed individually. The composition of the simple stretchable objects results in a complicated object. The flow chart of the reconstruction method is shown in Fig. 8.3. In addition to the 'simple' stretchable objects, some 3D objects can be considered as special stretchable objects. For example, prism and truncated cones can be reconstructed using the stretch method. One noted limitation of the stretch method is that it cannot be used to reconstruct objects with spherical surfaces. However, spherical objects are not encountered in the structural components in the building drawings; only the upholstering of the building contains such objects.

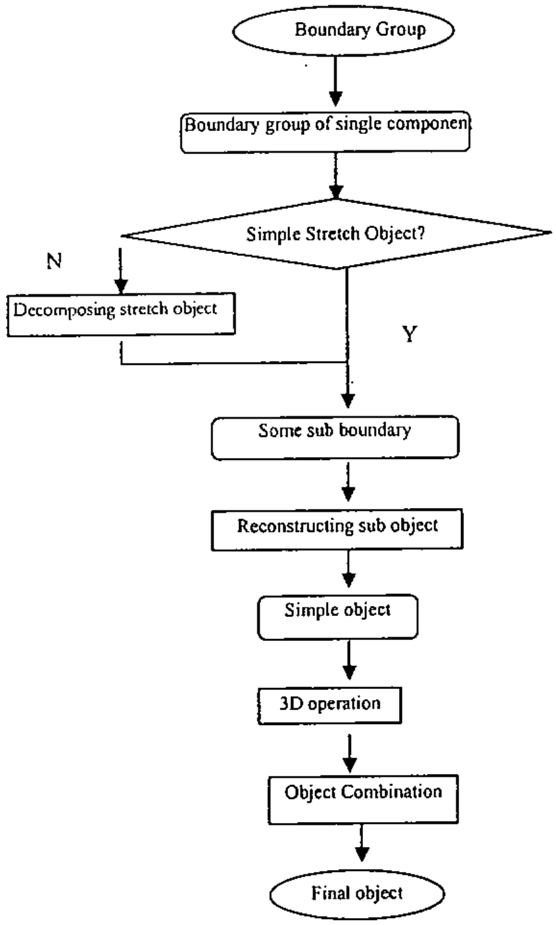


Figure 8.3 3D Reconstruction Method Based on Stretch of Objects

8.2.1 Automatic Regeneration of the Absent Projection

In building drawings, some standard parts are often represented by their code names, with the graphical details omitted from the drawings. In order to obtain the complete 3D components, the absent parts have to be automatically regenerated after analyzing the existing projection boundaries and the code names (symbols and texts etc.). The regeneration process is controlled by the following two rules:

(1) If the missing part is a column or slab, and only the top projection exists, then draw the circumscribed rectangle R(out) of the top projection. The front projection is

regenerated as a rectangle and its length is the length of the R(out), and the width of it is the height of the column (the data is obtained from the text information). The section projection is regenerated as another rectangle and the length of it is the width of the R(out), and the width is the height of the column.

(2) If the missing part is a beam or wall, and there exist two projections, then project the two projection boundaries to the third view, and the projections are two lines. The third projection is regenerated as a rectangle and the length of it is the length of one of the lines; the width is the length of the other line.

8.2.2 Decomposition of Complex Projections

If there are holes/voids in an object, the holes are shown as closed boundaries in the projections. The holes can be separated from the projections. For example, let us assume that there is a hole O(f) in object F(f), then the following procedure can separate the hole from the object.

(1) Project the O(f) into F(t), and the projection is a line L. From each end point draw a line that is orthogonal with F(f), then the two lines divide F(t) into three regions. The region that includes the line L is called the projection region (PR) of O(f) in F(t). The part of the boundary that in the PR is O(f)'s projection in F(t), called O(t).

(2) The O(f)'s projection in F(s) that called O(s) is obtained as in (1).

(3) {O(t), O(f), O(s)} is the projection group of the hole O. A hole flag is added to the projection group.

(4) Remove the O(f) from the F(t).

After identifying all holes' projection group from the main projection group, the projection group for the holes is removed from the object which can be reconstructed using the stretch method.

8.2.3 Actual Construction Component Reconstruction

After the decomposition and reconstruction of individual stretchable objects, these objects have to be recomposed to form the actual construction component. This is achieved in a two-step process: (1) aggregation of simple objects and (2) removal of false objects.

For example, if a construction component is composed of three objects, i.e. two true objects and one false object as shown in Fig 8. 4. A false object is a void that needs to be deducted from a true object.

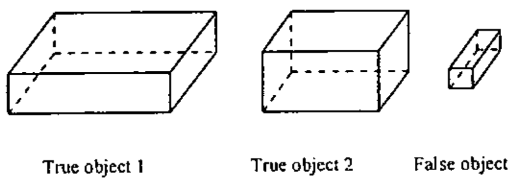


Fig. 8.4: Decomposed and Reconstructed 3D Objects of a Construction

Because all the objects are reconstructed using the stretch method in the same coordinate system, the objects are simply aggregated and 3D operations can be applied directly to obtain the actual construction component. Specifically, all the true objects are integrated by the 3D plus operation to obtain a temporary object. Then, using the 3D minus operation, the temporary object is reduced by the volumes of the false objects. The actual construction component obtained is shown in Fig. 8.5.

The reconstructed 3D objects of adjacent components based on the stretch method often have inter-penetrations at the joints. As a result, excrescent parts occur which should be removed.

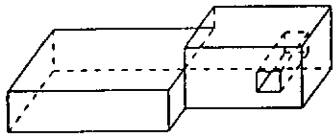


Fig 8.5: Actual Construction Component Resulted from Recomposition

In order to remove the excrescent parts, collision detection technique is employed. The collision detection technique used in this study is a simplified version of the Null Set Detection method reported in Cameron (1990), which searches for evidence of non-nullity, indicating the extrusions overlap and so that the collision of objects. As in the presented study, motions of objects are not considered. The simplified Null Set Detection operates in two stages:

- A divide-and-conquer stage, whereby the problem is dynamically split into a number of simpler problems to reduce the computational complexity;
- A generate-and-test stage, at which the exact geometry of the problem is considered.

The divide-and-conquer stage firstly mathematically formulates the collision detection problem as follows. Object *M* has center at (x1,y1,z1) and is shaped by a set of points Pi, {i=1, ,k}; Object *N* has center at (x2,y2,z2) and is shaped by a set of points Pj {j=1, ,l}. The task is to identify any non-nullity between objects *M* and *N*.

In order to maximize the processing efficiency, the divide-and-conquer stage classify the points of the objects into three categories. The first category includes the points on the surfaces facing the adjacent object. The second category includes the points on the surfaces facing the opposite direction of the adjacent object. The third category include the rest of the points.

In the generate-and-test stage, each point of Object M in the first category is tested to see if there is intersection set with Object N. If there is no intersection, then points in the third category are also tested one by one until all the points in the third category are tested. As this process is an exhaustive search, the complexity is not worse than $O(n^2)$.

If no evidence of non-nullity is discovered, then the two objects are not collided. Otherwise, collisions are detected, and necessary revisions and adjustments are needed to modify the dimensions of the objects.

8.3 Identifying Dimensional Inconsistencies Through Collision Detection

8.3.1 Identifying Dimensional Errors Between Different Projections

The 3D boundary information of a building components are construction from projections of the components in three orthogonal 2D views including the components' top view F(t), front view F(f), and section view F(s). So the dimension of each boundary can be achieved from two component's view. It is obvious that discrepancy exist if the dimensional information of same boundary from different views has collisions.

8.3.2 Grid System Construction

In order to determine whether there are any conflicts between building components the grid reference network of each floor should be recognized. Thus the accuracy of relationship of building elements can be maintained. The grid system is a rectangular excavation or sampling units lay over a site by grid elements. The recognition of grid element is controlled by the following rules:

(1) A grid element contains three parts: Grid Mark, Grid Line and Grid Circle.

(2) A Grid Mark is a string, its length is no more than 3 and each character of the Grid Mark belongs to the set of normal 26 christcross-row numbers.

(3) The Grid Line is the line that connects the Grid Circle

(4) The Grid Circle is the circle that has a Grid Mark inside.

8.3.3 Checking Completeness and Correctness of Dimension Sets

A perfect dimension check system should both completeness and correctness of dimensions. Completeness means that any geometric edge of an object should have correspond dimension in the projection direction and correctness can be called none-conflict. It means that for any edge if have several dimensions in the projection direction, these dimension do not conflict with each other.

In order to check the completeness and correctness of dimensions, a graph is constructed based on the information of dimensions. The dimension points and dimension lines are the nodes and edges of the graph. Then depth first search (DFS) on the graph is performed, and all the loops in the graph are achieved.

8.3.4 Identifying Dimensional Inconsistencies through Examining Reconstructed 3D Models

The results of the reconstruction process are a set of simple stretchable objects as shown in Fig. 8.6. Inconsistent dimensions between 2D orthographic views will result in irregular 3D objects which are non-existent in building and construction engineering.

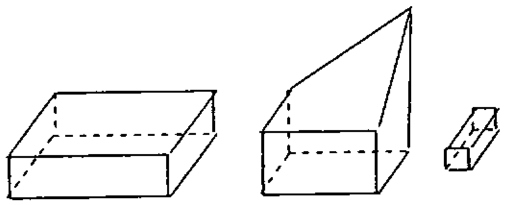


Fig 8.6 Reconstructing 3D Objects by Stretch Method

A visual examination of the reconstructed 3D models by an experienced engineer conveniently leads to the identification of the irregular objects. This in turn assists in spotting the dimensional inconsistencies that have caused the irregularities or abnormalities in the objects. For example, the irregular shape of Object 2 in Fig. 4 is suspicious and hence alerts the engineer. A request is then raised to the designer who eventually confirms the object being cubic following acknowledging a dimensional inconsistency in the drawings.

8.4 Conclusions

Dimensional inconsistencies are a major source of design errors which often lead to rework. In this chapter, the use of 3D reconstruction and collision detection techniques is presented to detect dimensional inconsistencies within a design objects as well as between objects. The 3D reconstruction technique can detect the dimensional inconsistencies within the same object, as inconsistent dimensions presented in three orthographic views result in an irregular/incomplete 3D model which can be easily spotted and corrected by designers. The collision detection technique allows the dimensional inconsistencies between objects to be detected and identified. It may be expected that by using these techniques, a substantial amount of design errors can be eliminated and the failure costs of rework minimized.

Chapter 9

CONCLUSIONS

9.1 Contributions of the Study

The research described within this thesis has achieved two main aims: (1) the incorporation of several techniques from the domain of engineering drawing understanding and interpretation into the construction industry, (2) the implementation of a system for automatic calculation quantities of steel reinforcement. The use of EDUI techniques provides a new solution for quantity surveying in the construction industry.

This research contributes to existing methods for EDUI in forms of new algorithms and their integrative implementation in a system. The algorithms overcome some bottleneck problems in EDUI such as symbol recognition, wall extraction and automatic recognition rules acquisition. Specifically, this study has made the following contributions:

- This study has developed a system to understand and interpret engineering drawings in construction and partially automated the quantity survey process. The developed system assimilates the quantity surveyors' professional knowledge and experience, in order to improve the efficiency and accuracy of the taking off process. The system is based on a set of redesigned, modified or new algorithms created specifically for the task described in this study, in order to achieve improved performance in the EDUI system.
- The algorithm developed for extracting the information relating to drawing version is the basis for automatic engineering drawing management. It

provides a flexible approach to controlling the development of different versions of engineering drawings.

- The symbol recognition order-adjusting algorithm developed in this research is to avoid the related errors in symbol recognition. The algorithm prioritizes the symbols to be recognized by using the similarity measurement of instances in a drawing under two recognition thresholds. Based on the error probability sequence, a suitable recognition sequence is performed and some of the related errors can be avoided without changing the recognition algorithm itself, so that both the accuracy and efficiency of the symbol recognition system can be improved.
- The wall extraction algorithm developed in this research is to detect wall information in architectural drawings. The results could be used for calculating the quantity of none structural wall. and to provide spatial and dimensional information for further understanding of structure drawings. For structural walls the information of reinforcement steel will be shown in details drawing while reinforcement steel of none structural wall will not be drawn. Only some typical details are shown for the calculation of the quantity. So it requires combining the information of wall shape recognized in architectural plans and the typical details information. The algorithm uses the door within a room as the seed segment to guide the recognition of wall. Compared with traditional approaches which only use parallel feature for wall recognition, the proposed method shows a considerable improvement both in accuracy and efficiency over existing methods.

- The automatic rule-acquisition algorithm developed in this research is to extract geometric features of architectural objects and convert the features into recognition rules. By adopting the function of automatic object feature extraction and rule acquisition presented in this study, rules in the recognition systems are continuously refined along with new objects introduced. The recognition ability is therefore improved without modifying the recognition algorithm. This mechanism will improve the self-adaptability, stability and applicability of application systems.
- Another contribution of this research is the development of a collision detection technique for construction elements in engineering drawings. The technique can reconstruct 3D models from three 2D orthographic views and automatically identify and minimize design errors. The collision detection technique allows the dimensional inconsistencies between objects to be detected and identified. By using these techniques, a substantial amount of design errors can be eliminated and the failure costs of rework minimized.

9.2 Evaluation of the Research (Limitations)

Although this research project has been accomplished with satisfied results, there are some limitations of the research which can be summarized as follows:

- The EDUI system cannot identify human errors and design errors in engineering drawings.
- Though the system has the ability of self-learning by automatic rule acquisition, it still needs a large number of rules to cover every possible condition. But as

the number of rules rises, the reasoning speed of the system tends to become slow.

- There is a lack of performance measures and experimental protocols. In other words, the population size of engineering drawings on which the developed algorithm 'works' is not well defined.

9.3 Recommendations for Further Research

Despite that many aspects of the construction management process have been computerized, quantity surveying is still largely a manual process. This situation needs to be improved so that human efforts can be released to focus on performing more creative jobs. This study presents a useful step towards automating the taking off process. To accomplish this entire task, the following research issues are to be tackled:

(1) Incorporation of more intelligent user interfaces (IUIs) into the EDUI algorithms to further increase the accuracy and speed.

Though a lot of EDUI algorithms have been presented, these algorithms can only deal with certain special cases very well. None of them can guarantee achieving perfect results due to data noise, lack of recognition ability or unique drawing errors. To solve these problems, IUIs are required which can be integrated with domain-specific drafting knowledge and inferring ability in the developed algorithms. IUIs are human-machine interfaces that aim to improve the efficiency, effectiveness, and naturalness of human-machine interaction by representing, reasoning, and acting on models of the user, domain, task, discourse, and media (e.g., graphics, natural language, gesture). Intelligent user interfaces are multifaceted, in purpose and nature, and they include capabilities for multimedia input analysis, multimedia presentation generation, and discourse and task models to personalize and enhance interaction. Two IUIs are potentially useful for the EDUI algorithms: Model-based Interfaces and Agent-based Interfaces. Model based interfaces separate applications into four layers: application actions, dialog control, style rules (specifications of presentation and behavior), and style program layer (primitive toolkit objects composed by style rules). In addition to supporting more declarative development, model based interfaces can draw upon the automated input analysis and output generation techniques. So model-based interface development environments promise automated design critique, refinement and implementation. Agents have increased prominence in applications recently. Examples of agents include search agents, desktop agents, collaborative filtering, and agents for intelligent distributed computing. Agents may assist in decreasing task complexity, bringing expertise to the user or simply providing a more natural environment with which to interact. Research in this area includes the use of agents to express system and discourse status via facial displays, multi-modal communication between animated computer agents, and developing standards and open architectures for building agent based multi-modal interfaces.

(2) Better text recognition algorithms

Textual information is the most important information in engineering drawings. Though there are some standards for presenting textual information in engineering drawings, there are still a lot of textual information not presented in standard formats in engineering drawings. To understand these textual information could help understanding the drawings enormously. This falls into the domain of natural language processing (NLP). To incorporate this capacity into the developed system will be another future research subject. Cross Language Retrieval, Data Mining and Entity Extraction are among the main NLP techniques that can be applied in EDUI systems. An increasingly polyglot world needs to have access to information on a subject no matter in what language it is written. So EDUI systems should have the ability to process texts in drawings no matter what language was used originally. Data Mining and Entity Extraction is a promising area of research in NLP. Good entity extraction is entirely an NLP-dependent process. It extracts names of people, places, and things from text and stores them, sometimes with other related information. NLP systems can go further and store the names together with indicators of what they are.

(3) Self-adaptive feature-based symbol recognition algorithms

Since there are significant differences in symbols used in different application domains, self-adaptive feature-based symbol recognition algorithms that are independent of application domains are desirable. These algorithms should be data-driven under the support of a symbol library which is dynamically formulated during the use of these algorithms. Decision tree, one of the techniques in machine learning, can be used. A decision tree is a tree whose internal nodes are input patterns and whose leaf nodes are categories of patterns. A decision tree assigns a class number to an input pattern by filtering the pattern down through the nodes in the tree. By adjusting the tree structure and attaching dynamic features to the nodes, a decision tree can have the self-adaptive ability used in different application domains.

(4) Powerful semantic analysis methods for 3D reconstruction and information extraction

Much attention has been paid to syntactic analysis and less to the semantic understanding of drawings in current studies. Since much data are lost or distorted in the process of converting 2D drawings to a 3D model, information obtained by analyzing the drawings may be not completed. There are numerous 3D reconstruction methods which are purely based on syntactic operations. The basic idea of these methods is to look for sets of consistent matching hypotheses between features extracted from three orthogonal projection views of an object. All these methods have in common two limitations (Tombre 1995): (1) the processed data have to be perfect. The methods assume that a clean idealized set of projection is provided. (2) These methods can only be applied to geometric part of the drawing. However as a technical document is not only set of geometric patterns but also a "linguistic" dimension, i.e. information represented by different symbolic means. Further study is needed to develop 3D reconstruction techniques which can be combined with other techniques such as symbol recognition, text understanding and spatial analysis at semantic level.

In summary, this study has developed a system for the understanding and interpretation of reinforcement drawings for the quantity surveyor in construction. The algorithms and components applied in the system have provided a good starting point for the practice of EDUI in construction as well as for further research in this area.

References

Ablameyko S. et al., (1993), Computer-Aided Cartographical System for Map Digitizing, In *Proceedings of 2nd ICDAR*, Tsukuba, Japan, 115-118

Ablameyko S. (1999) From Recognized Engineering Drawings to 3D Object Reconstruction, *GREC'99 3RD IAPR Workshop on Graphics Recognition*, Jaipur, India, 313-319.

Ah-Soon, C. (1997) A Constraint Network for Symbol Detection in Architectural Drawings, *GREC'97 Second IAPR Workshop on Graphics Recognition*, Nancy, France, 80-90.

Ah-Soon, C. (1997) Symbol Detection in Architectural Drawings, *GREC'97 Second IAPR Workshop on Graphics Recognition*, Nancy, France, 280-286.

Ah-Soon, C., Tombre, K. (1997) Variations on the Analysis of Architectural Drawings, *Proceedings of the 4th International Conference Document Analysis and Recognition (ICDAR '97)*, Ulm, Germany, 18-20.

Ah-Soon, C. and Tombre, K. (1998) Network-Based Recognition of Architectural Symbols, *In Advances in Pattern Recognition (Proceedings of Joint IAPR Workshops SSPR'98 and SPR'98), Lecture Notes in Computer Science*, 1998. Vol. 1451, 252-261.

Ah-Soon, C. (1998) A Constraint Network for Symbol Detection in Architectural Drawings, *Graphics Recognition--Algorithms and Systems*, vol. 1389 of Lecture Notes in Computer Science, 80-90.

Aldefeld, B. (1983) On automatic recognition of 3D structures from 2D representations. *Computer Aided Design*, 15(2), 56-64.

Aoki Y. (1996) A prototype System for interpreting Hand-Sketched Floor Plans, *In Proceedings of the 13th International Conference on Pattern Recognition*, Vienna, Austria, vol. 3, 745-751.

Balachandran, M., Rosenman, M.A. and Gero, J.S. (1991) A knowledge –based approach to the automatic verification of designs from CAD database, *Artificial Intelligence in Design*, 757-781.

Benjamin, D. (1998) The Use of High- level Knowledge for Enhanced Entry of Engineering Drawings, *Proc. 9th ICPR*, Milano, Italy, 119-124.

Bijl, A. (1986) AI in architectural CAD, *Edinburgh Computer Aided Architectural Design Report.*

Boatto L. et al., (1992), An Interpretation System for land Register Maps, *IEEE Computer*, Vol.25(7), 25-33

Bouma, W and Vanecek, J. G. (1991) Collision detection and analysis in a physically based simulation. *Proceedings of Second Eurographics Workshop on Animation and Simulation*. Vienna, Austria, 191-203.

Bridgewater, C and Griffin M. (1993) Application of virtual reality to computer-aided building design. In JJ Connor, S Hernandez, T K S Murthy and H Power (eds) *Visualization and Intelligent Design in Engineering and Architecture*, Elsevier Science Publishers, USA. 213-218.

Buchanan, B., Sutherland G., and Feigenbaum E. (1969) *HEURISTIC DENDRAL: A program for Generating Explanatory Hypotheses in Organic Chemistry. Machine intelligence 4.*, Edinburgh University Press.

Canny, J (1986)Collision detection for moving polyhedron, *IEEE Tranactions on Pattern Analysis and Machine Intelligence*. 8(2), 200-209.

Cao, Y. (1999) Automatic Interpretation of Structural Drawings, *Proceedings of Third Internation Workshop on Graphics Recognition*, Jaipur, India, 298-305.

Cao, Y., Wang, S.H. and Li, H. (2002) Automatic Recognition of Tables in Construction Tender Documents, *Automation in Construction*, 11 (5), 573-584.

Ceri S. and Manthey R. (1993) Consolidated specification of Chimera, the conceptual interface of IDEA, *Technical Report IDEA.DD.2P.004*, ESPRIT Project 6333 IDEA.

Chadwick, M. and Hannah, J.A. (1986) *Expert Systems for Personal Computers - An Introduction to Artificial Intelligence*, Sigma Press, Wilmslow, England, Chapter 2: Principles of Rule-based Expert Systems, 9-29.

Chandran S., Kasturi R. (1993) Structure Recognition of Tabulated Data, *In Proceedings of 2th International Conference on Document Analysis and Recognition*, Tsukuba Science City, Japan, 516-519.

Chhabra A.K., (1997), Graphic Symbol Recognition: An Overview, *GREC'97 Second IAPR Workshop on Graphics Recognition*, Nancy, France, 68-79.

Clove L. et al. (1991), Knowledge-Based Systems in Engineering, *In Clive L. Dym & Raymond E. Levitt* (eds)

Collin S. et al. (1994) Syntactic analysis of technical drawing dimensions, *International Journal of Pattern Recognition and Artificial Intelligence*, 8(5), 1131-1148.

Coyne, R. and Flemming, U. (1990) Planning in design synthesis: Abstraction Based LOOS, *Artificial Intelligence in Engineering 1: Design*, Springer, 91-111.

Coyne, R. (1991). *ABLOOS: An Evolving Hierarchical Design Framework.* Ph.D. Thesis, Department of Architecture, Carnegie Mellon University, Pittsburgh, PA.

February 1991. Also available as a Technical Report, *EDRC-02-15-91*, Engineering Design Research Center, CMU.

Coyne, R., Flemming, U., Piela, P. and Woodbury, r. (1993) Behavior modeling in design system development, *CAAD Futures '93*. Proceedings of Fifth International Conference on Computer-Aided Architectural Design Futures, Pittsburgh(USA), 7-10 July. U. Flemming and S. Van Wyk, eds., North-Holland, NY. 335-354.

Coyne, R. and Subrahmanian, E. (1993) Computer supported creative design: a pragmatic approach, *Modeling Creativity and Knowledge-Based Creative Design*. J. Gero and M. Maher, eds., Lawrence Erlbaum, Hillsdale, NJ. 295-327.

Cunningham, J. and Dixon, J.R. (1998) Designing with Features: The Origin of Features, *ASME Computers in Engineering Conference*, San Francisco, USA, 237-243.

Della, V. A. and Schettini, R. (1994) Graphic symbol recognition using a signature technique, *Proc. 12th IAPR International Conference of Pattern Recognition (Conf. B: Pattern Recognition and Neural Networks)*, IEEE Computer Society Press, USA, 533-535.

Dix, A.J. and Miles, V.C. (1992) *Version control for asynchronous group work*, YCS 181, Department of Computer Science, University of York.

Donald, B R (1987) A search algorithm for motion planning with six degree of freedom, *Artificial Intelligence*, 31(3), 295-353.

Dori D., (1989), A Syntactic/Geometric Approach to Recognition of Dimensions in Engineering Machine Drawings. *Computer Vision, Graphics, and Image Processing*, Vol.47, 271-291.

Dori, D. (1992) Dimensioning analysis: toward automatic understanding of engineering drawings, *Communications of the ACM*, 35(10), 92-103.

Dori, D. and Tombre K. (1994) From engineering drawing to 3-D CAD models: Are we ready now?, *Computer-Aided Design*, 29(4), 243-254..

Dori, D. (1995) Representing pattern recognition-embedded systems through object-process diagrams:the case of the machine drawing understanding system, *Pattern Recognition Letters* ,16, 377-384.

Dori, D. and Liu W.Y. (1999) Automated CAD Conversion with the machine drawing understanding system: Concepts, Algorithms, and performance. *IEEE Transactions on Systems, Man, and Cybernetics*, Part A: systems and humans, 29(4), 411-416.

Dosch P., Masini G, (1999) Reconstruction of 3D structure of a building from the 2D drawing of its floors, ICDAR'99, Proceedings of Fifth International Conference on Document Analysis and Recognition, Bangalore, India, 487-90.

Dosch P., Ah-Soon, C., Masini, G., Sánchez, G. and Tombre K. Design of an integrated environment for the automated analysis of architectural drawings. In S.-W. Lee and Y. Nakano, editors, *Document Analysis Systems: Theory and Practice. Selected papers from DAS'98*, Springer Verlag, 1999. Volume 1655 of Lecture Notes in Computer Science, 295-309.

Dym, C.L., Hanchey, R.P., Delis, E.A. and Gonic S. (1988) A knowledge based system for automated architectural code checking, *Computer Aided Design*, 20(3), 137-145.

Fahn C.S. (1988) A Topology-based Compoment Extractor for Understanding Electronic Circuit Diagrams, *Computer Vision, Graphics, and Image Processing*, 44, 119-138.

Falk G. (1972) Interpretation of imperfect line data as three-dimensional scene, *Artificial Intelligence*, 3(2), 101-144.

Fenves, S J and Baker N C (1988) Grammar for functional and spatial Reasoning in design. *Proceedings of fifth Conference on Computing in Civil Engineering*, San Francisco, USA, 819-828.

Fenves, J. S., Rivard, H., Gomez, N. and Chiou, S.-C. (1995). Conceptual structural design in SEED, *in Journal of Architectural Engineering, ASCE*, 1(4): 187-194.

Flemming, U., Coyne, R., Glavin, T. and Rychener, M. (1988) A Generative Expert System for the Design of Building Layouts - Version 2. *Artificial Intelligence in Engineering: Design.* Proceedings of the Third International Conference, Palo Alto, CA., J. Gero ed., New York: Elsevier. 445-464.

Flemming, U. (1989) More on the representation and generation of loosely-packed arrangements of rectangles, *Environment and Planning B. Planning and Design*, 16, 327-359.

Flemming, U. Woodbury, R. and Coyne, R. (1993) SEED: A software environment to support the early phases in building design, *Proceedings of ARECDAO 93.* Barcelona, Spain, 61-85.

Fu, K.S. (1982). *Syntactic Pattern Recognition and Application*, Prentice-Hall, Englewood, Cliffs, N.J.

Gakuhito, H. and Kataoka, M. (1999) Development of a Drawing Version Control System, used to support alternative designs and ensure the integrity of external references, *CONCURRENT ENGINEERING INCONSTRUCTION: CHALLENGES FOR THE NEW MILLENNIUM*, CIB Publication 236,161-166.

Gloger, J. M., Pasternak, B., Sprengel, R., Gabrielides, G., Luth, N. and Timmermann M. (1992) An AI-Aided System for the Conversion of Paper-Based Technical Drawings into a CAD Format, Proceedings of the IAPR Workshop on Machine Vision Applications - MVA '92, Tokyo, Japan.

Hahn J (1988) Realistic animation of rigid bodies, *Computer Graphics*, 22(4), 299-308.

Han C.S. et al., (1997) Making Automated Building Code Checking a Reality, *Facility Management Journal*, 22-28.

Hammarlund, Y., and Josephson, P. E. (1991). Sources of Quality Failures in Building. *Proceedings of the European Symposium on Management, Quality and Economics in Housing and Other Building Sectors.* Lisbon, Portugal, September 30th - October 4th, 671-679.

Henderson, M.R. (1984) Extraction of feature information from three dimensional CAD data, PH.D. Thesis, Purdue University.

Houten V. et al. (1989) Part, A CAPP system with a flexible architecture, *In second CIRP Workshop on CAPP* (Hannover).

Huang, M, Fu, S., and Pridomore, T. P. (1997) Interpreting images of architectural drawing for building cost estimation, *6th IEEE International Conference on Image Processing and its Applications, 15-17 Juli 1997, Conference Publication* No. 442, 126-130

Huang Kou-Yuan, (2002), Syntactic Pattern Recognition for Seismic Oil Exploration, World Scientific.

Huffman, D. A. (1971) Impossible objects as nonsense sentences, *Machine Intelligence,* Eds. B. Meltzer and D. Michie, Edinburgh University Press, Edinburgh, 6, 259-233.

Idesawa, M (1972) A system to generate a solid figure from three views, *Transaction of the Japan Society of Mechanical Engineers*, 38(310), 1267-1276.

Itonori, K. (1993) Table Structure Recognition based on Textblock Arrangement and Ruled Line Position, *In Proceedings of 2^{th} International Conference on Document Analysis and Recognition*, Tsukuba Science City, Japan, 765-768.

Joseph S.H. (1992) Knowledge-Directed Interpretation of Mechanical Engineering Drawings, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(9), 928-940.

Joshi, S. and Chang, T.C. (1988) Graph-based heuristic for recognition of machined features from a 3D solid model, Computer Aided Design, 20(2), 58-66.

Lladós, J., López-Krahe, J. and Marti, E. (1996) Hand Drawn Document Understanding Using the Straight Line Hough Transform and Graph Matching, In Proceedings of the 13^{th} International Conference on Pattern Recognition, Vienna, Austria, 2, 497-501.

Lladós, J. et al. (1998) A String Based Method to Recognize Symbols and Structural Textures in Architectural Plans, *Graphics Recognition--Algorithms and Systems*, vol. 1389 of Lecture Notes in Computer Science, 91-103.

Lladós, J. and Martí, E. (1999) A graph-edit algorithm for hand-drawn graphical document recognition and their automatic introduction into CAD systems, *Machine Graphics & Vision*, 8(2), 195-211.

Kernighan, B. K. and Christopher J. V.W., (1996) Extracting Geometric Information from Architectural Drawings, *Workshop on Applied Computational Geometry*, Philadelphia (USA), 167-176.

Kitamura, Y, Smith, A, Takemura H and Kisdoc, F (2000) Parallel algorithms for real-time colliding face detection. Proceedings of IEEE International Workshop on robot and Human Communication (ROMAN'96), Tsukuba, Japan, 834-846.

Koutamanis, A. (1989) Recognition of Locations in Architectural Plans, *In Proceedings of 6th Scandinavian Conference on Image Analysis*, Oulu, Finland, 800-803.

Koutamains, A. and Mitossi, V. (1992) Automated Recognition of Architecture Drawings, *In Proceedings of 11th International Conference on Pattern Recognition*, Den Haag ,Netherlands, 660-663.

Kyprianou, L.K. (1980) Shape classification in computer-aided design, Ph.D thesis, Cambridge Univerisity, U.K.

Kwon, Y.B. (2003) Main Wall Recognition of Architectural Drawings using Dimension Extension Line, *GREC'2003 Fifth IAPR Workshop on Graphics Recognition*, Barcelona, Spain, 145-156.

Larry B. (1999) Interpretation of Wiring Diagrams for Aerospace Information Systems, *GREC'99 3RD IAPR Workshop on Graphics Recognition*, Jaipur, India, 307-311.

Lee, M.C. (1991). IPDA - an expert system for debugging Prolog programs, *International Journal of Artificial Intelligence in Education*, 2(4), 91-101.

Levitt, R E, Kunz J C and Kartum N A (1987) Using artificial intelligence techniques for automated planning and scheduling. *Proceedings of Fourth International Symposium on Robotics and Artificial Intelligence in Building Construction*, Haifa, Israel, 27-52.

Lewis and Sequin C. (1998) Generation of 3D building models from 2D architecture plans, *Computer Aided Design*, 30(10), 765-779.

Li, H and Love, P (1998) Visualization of Building Interior Design to Reduce Rework, *Proceedings of International Conference on Information Visualization*, London, UK, 187-191.

Li, H (1998) The Use of Visual Simulation in Construction Technology Education. Computer Applications in Engineering Education. 6(4), 217-222.

Lin M and Canny J (1998) Efficient collision detection for animation, *Proceedings of the Fourth Eurographics Workshop on Animation and Simulation*. Cambridge, UK, 205-213.

Liu, S. X., Hu, S. M., Chen Y. J. and Sun J. G. (2001) Reconstruction of curved solids from engineering drawings. *Computer-Adied Design*, 33, 1059-1072.

Liu W. and Dori D. (1998) Genericity in Graphics Recognition Algorithms, *Graphics Recognition -- Algorithms and Systems, eds. K. Tombre and A. Chhabra, Lecture Notes in Computer Science*, Springer, Vol. 1389, 9-20.

Lo S.M. et al. (2000), A View on the Intelligent Understanding of CAD Building Plans, *The Hong Kong Surveyor*, Vol. 11(4), 4-8.

Iwata K. et al., (1988), Recognition System for Three-View Mechanical Drawings, In *Proceedings of 9th ICPR*, Cambridge (UK), 240-249.

Maderlechner, G., et al. (1986) Knowledge-based pattern in disturbed line drawings, *IEE Proc. 2nd International Conference on Image Processing and Application*, 139-143.

Maderlechner, G., et al. (1988) Classification and Modeling of Line Drawings, *In Proceedings. 9th ICPR*, Milano, Italy, 678-680.

Meeran S and Pratt, M J (1993) Automated recognition of 3D solid objects from 2D orthographic views. Pattern Recognition, 21(5), 439-449.

Messmer, B. and Bunke, H. (1996) Automatic learning and recognition of graphical symbols in engineering drawings, In R. Kasturi and K. Tombre, editors, *Graphics Recognition: Methods and Applications, Selected Papers from First International Worksshop on Graphics Recognition*, 1995, Vol. LNCS 1072, Spinger, Berlin, 123-134 .

Mondal, P, Love, P, Smith, J and Li, H (2000) Modeling the dynamics of design error induced rework in construction projects. Construction Management and Economics. 18(5), 567-574.

Moore, M and Wilhelms J (1988) Collision detection and response for computer animation, *Computer Graphics*, 22(4), 289-298.

Nagendra, I V and Gujar U G (1988) 3D objects from 2D orthographic views, a survey. *Computer and Graphics*, 12(1), 111-114.

Niyogi, D. and Srihari, S. N. (1986) A rule-based system for document understanding, *Proc. AAAI*, Los Altos, Canada, 789-793.

Nnaji, B.O. and Kang, T. (1990) Interpretation of CAD models through neural geometric knowledge, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 4(1), 15-45.

Ogier J. M., et al., (1993), Attributes Extraction for French Map Interpretation, In *Proceedings of 2nd ICDAR,* Tsukuba, Japan, 672-675

Okazaki, A., Kondo, T., Mori, K. and Tsunekawa S. (1988) Kawamoto, An Automatic Circuit Diagram Reader with Loop-Structure-Based Symbol Recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(3), 331-341.

Okazaki S. and Tsuji, Y. (1988) Knowledge-based Approach for Adaptive Recognition of Drawings, *Proc. Pattern Recognition, Fourth Int'l Conf*, 627-632.

Pasternak, B., Gabrielides, G. and Sprengel, R. (1992) WIZ-Design and Development of a Prototype for Knowledge-Based Interpretation of Technical Drawings. *Labor fürKünstliche Intelligenz Technical Report* LKI-M-92/1.

Pasternak, B. and Neumann, B. (1993) Adaptable Drawing Interpretation Using Objecte-Oriented and Constraint-Based Graphic Specification, *Proceedings of the second International Conference Document Analysis and Recognition (ICDAR '93)*, Tsukuba, Japan, 359-364.

Pasternak, B. (1995) The Role Taxonomy in Drawing Interpretation, *Proceedings of the 3rd International Conference Document Analysis and Recognition (ICDAR '95)*, Vol. 2, Montréal, Canada, 799-802.

Pollock, A.J., Donegan, H.A. and Taylor, I.R. (1994) Interfacing Architectural CAD Data with KBS Technology, *Mathematical Computer. Modelling*, 19(9), 17-21.

Prabhu B.S., Pande S.S. (1999) Intelligent interpretation of CADD drawings, *Computers & Graphic*, 1999(23), 25-44.

Qiang Ji and Michael M. Marefat, (1997) Machine interpretation of CAD data fro manufacturing applications, *ACM Computing Surveys*, 24(3), 264-311.

Qi Z. (1999) Application of Product Modelling Technology in Building Plan Checking System, *2nd International Conference on Construction Industry Development*, Singapore.

Rangachar Kasturi, Sing T. Bow et al. (1990) A System for Interpretation of Line Drawings, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10), 978-992.

Reeth, V., Welter, F. and Flerackers, E (1990) A new parallel anti-aliasing method for Z-buffer algorithms. *Proceedings of Computer Graphics International'90*, NY, USA, 241-254.

Reiher, E., Said, F., Y. Li and Suen C. Y. (1996) Map Symbol Recognition Using Directed Hausdorff Distance and a Neural Network Classifier, *Proceedings of International Congress of Photogrammetry and Remote Sensing*, , Vienna, Vol. XXXI, Part B3, 680-685.

Retik, A, Warszawski A and Banai A (1990) The use of computer graphics as a scheduling tool, *Building and Environment*, 25(2), 133-142.

Ryall, K. and Shieber, S. (1995) Semi-Automatic Delineation of Regions on Floor Plans, *In Proceedings of Thired International Conference on Document Analysis and Recognition*, Montreal, Canada, 964-969.

Sakurai, H and Gossard D. C (1983) Solid model input through orthographic views. *ACM/SIGGRAPH*, 17(3), 243-252.

Samet, H. and Soffer, A. (1994) A legend-driven geographic symbol recognition system, *In Proceedings of the 12th International Conference on Pattern Recognition*, Jerusalem, Israel, Vol II, 350-355.

Sánchez, G., Valveny, E. Lladós, J., Martí, E., Ramos, O., Lozano, N., Mas, J. (2003) A system for Virtual Prototyping of Architectural Projects, *GREC'2003 Fifth IAPR Workshop on Graphics Recognition*, Barcelona, Spain, 65-74.

Shamilian, J.H., Baird, H.S. and Wood, T.L. (1997) A Retargetable Table Reader, *In Proceedings of 4th International Conference on Document Analysis and Recognition*, Ulm, Germany, 158-163.

Shin B S and Shin Y G (1998) Fast 3D solid model reconstruction from orthographic views, *Computer-Aided Design*, 30(1), 63-76.

Shortliffe, E. (1976) *Computer-Based Medical Consultations: MYCIN*, American Elsevier Co., New York.

Shum S P, Lau W S, Yuen M F and Yu K M (1997) Solid reconstruction from orthographic opaque views using incremental extrusion. *Computer & Graphics*, 29(6), 787-800.

Siller, T. J. (2000) Expert systems in geotechnical and environmental engineering, In Kim (ed) *Survey of the Sate-of-the-Art Expert/Knowledge Based Systems in Civil Engineering*, Champaign, 101-120.

Slade, S. (1991) Case-Based Reasoning: A Research Paradigm, *AI MAGAZINE*, 12, 42-55.

So, C., Bacju, G. and Sun, H. (1998) Reconstruction of 3D virtual buildings from 2D architectural floor plans, *Proceedings of the ACM symposium on Virtual reality software and technology*, Taipei, China, 17-23.

Song Z-L, Guo C-Y and Wang H-L, (2002) A room search method in architectural plan and its application, *Journal of Computer-Aided Design and Computer Graphics*, 12(3), 216-219.

S. Suzuki et al., (1990), Map Recognition Input System. *Pattern Recognition*, Vol.23(8), 919-933

Tikerpuu J. and Ullman D.G. (1988) General feature-based frame representation for describing mechanical engineering design developed from empirical data, *Computers in Engineering conference (CIE)*, 245-253.

Tombre, K. (1995) Graphics recognition – general context and challenges, *Pattern Recognition Letters*, 16, 883-891.

Tombre, K. (1997) Analysis of engineering drawings: State of the art and challenges, *GREC'97 Second IAPR Workshop on Graphics Recognition*, Nancy, France, 54-61.

Tombre, K. and Dori D. (1997) Interpretation of Engineering Drawings. *Handbook of Character Recognition and Document Image Analysis*, World Scientific,1997, In Bunke and Wang, Chapter 17, 457-484.

Tombre K. (1998) Ten Years of Research in the Analysis of Graphics Documents : Achievements and Open Problems. In *10th Portuguese Conference on Pattern Recognition*, Lisboa, Portugal, 11-17, March 1998.

Tzafestas C and Coiffet P. (2000) Real-time collision detection using spherical octrees: VR application, *Proceedings of IEEE International Workshop on Robot and Human Communication (ROMAN'96)*, Tsukuba, Japan, 834-846.

Vaxivire P. Tombre K. (1994) Knowledge Organization and Interpretation Process in Engineering Drawing Interpretation, Proc IAPR Workshop on Document Analysis Systems, Kaiserslautern, Germany, Oct. 1994, 313-321.

Walts, D. L. (1975) Understanding line drawings of scenes with shadows, *The psychology of Computer Vision,* Ed. P. H. Winston. McGraw-Hill, New York, 4, 458-469.

Wang, E. (1992) Using automatic feature recognition to interface CAD to CAPP, In proceedings of ASME Conference on Computers in Engineering, Vol. 1, 215-231.

Wang, S., Cao, Y., Yang R.Y., Cai, S.J. (2002) A Rule-Based System for Automated Quantity Determinations of Steel Reinforcement in Concrete Structures, *Automation in Construction*, 11, 607-616.

Watson, L., (1997) *Applying Case-Based Reasoning*, Morgan Kaufmann Publishers, Inc.

Wesley M and Markowsky G (1981) Fleshing out projection, *IBM Journal of Research Development, 25(6),* 934-954.

Wong W.C. et al., (2001), Singapore Automated Plan Checking System From Dread to Reality, *Interop AEC+fm 2001 Conference*, Sydeny, Australian, Oct 29-30, 2001

Yang R.Y. et al. (2000) Concrete Reinforcement Quantity Survey System Based on Graphics Understanding, Journal of Computer-Aided Design & Computer Graphics, 12(9), 688-694.

Yessios, C., Parent, R., Brown, W. and Terzidis, C. (1989) Knowledge-Aided Architectural Problem Solving and Design, *NSF Report DMC-8609893*, 12.

Yeung, D.S., Wang, X.Z. (2002) Improving Performance of Similarity-Based Clustering by Feature Weight Learning, *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 24(4), 556-561.

You, C. F. and Yang, S. S. (1996) Reconstruction of curvilinear manifold objects from orthographic views, *Computers. & Graphics*, 20(2), 275-293.

Yu Y.H., Samal, A. (1997) A System for Recognizing a Large Class of Engineering Drawings, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(8), 868-890.

Zhi G.S., Lo S.M., Fang Z. (2003) A Graph-based Algorithm for Extracting Units and Loops from Architectural Floor Plans for a Building Evacuation Model, *Computer-Aided Design*, 35,1-14.