An Object Models Development Tool To Support Image Understanding System *

Ke Chen
Zhongrong Li
Dept. of Computer Science and Engineering
Harbin Institute of Technology
Harbin 150006, P.R. China

Abstract

It is well known that a knowledge base is one of the most important parts in image understanding system, and object models are indispensable to the knowledge base. In general, there are two types of representations on object models, viz., geometric structure models and relational structure models. However, there is a big gap between them. In this paper a development tool is presented to bridge the gap and support a special image understanding prototype based on RBC theory, which is being developed at HIT. Some problems on this research

Keywords:
Image understanding, object models, visual language, spatial parsing, development tool.

1. Introduction

Computer vision has been an interesting theme in the field of Artificial Intelligence for a several years. The purpose of computer vision is to make computer capable of understanding environments from visual information. A primary objective in computer vision research is to construct image understanding systems which can analyze image based on object models. Usually, an image understanding system analyzes image by constructing interpretations in terms of the object models given to knowledge base of the image understanding system. Here, interpretation refers to the mapping between objects in the object model and image structure in the image.

As an important component of knowledge base, object models in question play a vital role in the porcess of image understnading. Hence, it becomes a big problem how to build up object models in the research of image understanding. In general, there are two types of repersentation on object models in an image understanding system[1], viz., geometric structure models and relational structure models. After investigating two types of object models, we found that many of relational structure models can be derived from their corresponding geometric structure models. For this reason, we design an object models development tool to build up relational structure models. As a result, this object models development tool will be employed to support an image understanding prototype based on RBC theory[2], which is a novel theory of human image understanding in cognitive psychology[3].

In this paper, we first introduce the research background, related theory and techniques. In section 3, we describe system overview on the development tool. This is followed by a discussion of some problems related to this research. Conclusions are presented in the final section.

2. Research Background and Related Theory & Techniques

2.1 Research Background

Recently a novel theory of human image understanding has been developed, called Recognition-By-Components(RBC). From the viewpoint of cognitive psychology, human beings can viewed from a novel orientation, or it is a novel exemplar, or its image is extensively degraded. Moreover, most often only a simple, brief fixation is all that is required to achieve quick and automatic understanding. RBC theory represents an object as an arrangement of simple generalized-cone volumes. The fundamental assumption of RBC is that a particular set of these convex components, called geons($N \leq 36$), can be derived from invariant properties of edges in a 2D image. If an arrangement of two or three geons can be recovered from the input, objects can be quickly recognitized even when they are occluded, rotated in depth,novel, extensively degraded, or embedded in a scene.

* This research was supported in part by National High Technology Foundation (86 3 Projects) Under Grant CS880332-015.

Using major viewpoints of RBC theory, we present an image understanding system to support aerial image understanding[4]. Usually, there are two types of object models in question stored in the knowledge base of the system. In order to build up those relational structure models, we have to derive them from their corresponding geometric strcture models by hand. This process is usually not efficient and time consuming. After investigating the process, we found that it is possible to models automatically by means of generalized icons theory and visual language techniques, for a relational structure model is completely dependent on the spatial arrangement of its corresponding geometric structure model. It is our objective of the research to develop an object models development so as to generate relational structure models according to those corresponding geometric structure models.

2.2 Related Theory and Techniques

Recently a generalized icons theory[5] has been presented by S.K. Cheng. Meanwhile, the techniques of visual programming language has also been highly developed and increasingly paid much attention to. Both of them supply some good ideas and efficient techniques for our research.

2.2.1 Generalized Icons Theory

Generalized icons theory points out that all pictures can be composed of some generalized icons, which are dual representations of objects consisting of a logical part and physical part. Generalized icons can be further classfied into object icons and process icons. A complex icon can be composed from other icons in the iconic system, which is a structured set of related icons, and therefore express a more complex visual concepts. Based on these assumptions, a formal specification of icon systems and a set of icon algebra operators are given to support visual language.

2.2.2 Visual Programming Techniques

The precious research shows that the visual programming techniques are related to visual languages. Visual languages, however, have large range and can be classified into four types: language supporting visual interaction, visual programming languages, viaual information processing language. and iconic visual information processing languages. In resent years,there appear a lot of reports on visual language and its application. a "compiler", which usually includes spatial parsing and symbolic result generating. Moreover, different types of visual language have different compilers, which can be developed according to particular processing approaches.

What we describe above only makes a fast survey of generlized icons theory and visual programming techniques. If readers are interested in these topics, they can find out them in references [5],[6] in detail.

3. System Overview

3.1 System Architecture

In section 2, we have introduced the research background. According architure shown in Fig.1. Based on RBC theory, we assump that an object models consist of some components, which will be defined in the next subsection in detail, in our image understanding prototype.

Monitor is the heart of the system. It monitors user interface and switches different states of the user interface in the light of user's choices. A state transition diagram is shown in Fig. 2. At the same time, it monitors other parts of the system so that these parts can work normally. In addition, it feedbacks processing results to users.

Under the control of monitor, other parts of the system play different role in the system, respectively. COMPONENTS MANAGEMENT can be employed to append/delete components and manage component dictionary. SYSTEM MAINTEMANCE can be employed to maintain MODEL DICTIONARY(MD). MODEL GENERATING is used to create or append a new object model to MD. In the following subsections, some design and implementation issues on these parts will be presented in detail.

3.2 User Interface

The objective of object models development tool is to supply an efficient interactive tool so that it is convenient to user to construct models. The user interface is designed, which centers on the fundamental objective from beginning to end.

In this system, an object model can be viewed as a spatial arrangement of constructing model layout by the interactive input/output devices. For this reason, a user interface spcification language presented. In order to obtain a friendly user interface, the screen is devided into three regions. We call them working region, component dictionary showing region and command input region, respectively, where working region is employed to make model layout. In command input region, users can use a keyboard or a mouse to select a COMMAND MENU. As one commands. In addition, the keyboard or mouse can be employed for users to make a model layout, switch states from one to another among three kinds sataes and define new components. In order to manage the system prefectly, a user interface specification language has been supplied for users. Using the specification language, a user can define models and components not existed in these dictionaries. With respect to components, we borrowed S.K. Cheng's generalized icons theory to define them. That is to say that every component ought to be composed of two parts, namely, a logical part and a physical part, usually denoted into (X_m, X_i) What the system preforms is to derive the logical part from the physical part. The following is the user interface specification language definition by BNF.

< Model> ::=
< Model Layout> < Model Meaning> < Model Name> < Model Type>

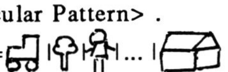
< Model Layout> ::=
Components Spatial Arrangement.

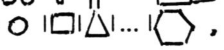
< Model Meaning> ::=
Model Description {Model Background Description}.

< Model Name> ::=
Character String | Arabic Numerals | Character String-Number.

< Component> ::=
< > |< Pm,Lm> .

< Pm> ::=
< > | < Standard Pattern> | < Ordinary Pattern> |< Particular Pattern> .

< Standard Pattern> ::=  .

< Ordinary Pattern> ::=  .

< Particular Pattern> ::= Icons defined by user himself.

< Lm> ::=
< > |< SP-Lm> |< OP-Lm> |< PP-Lm> .

< SP-Lm> ::=
< Attribute List Predefined> .

< OP-Lm> ::=
< Attribute List> .

< PP-Lm> ::=
< Attribute List> .

< Attribute List Predefined> ::= < Attribute List> ← -Contents.

< Attribure List> ::= < > | Name;No.;{Special Attribute Descrptions};{Key Tag}.

< Model Type> ::=
Semantic Nets | Attributed Relation Graph | ... | Framework.

It should be pointed out that those alternative items(e.g. Special Attribute Description, Key tag) are design for the special purpose of developing our image understanding prototype based on RBC theory.

3.3 Spatial Parsing and Model Generating

Model Generating (MG) is the most important and essential function of the development tool. It is its goal to generate a symbolic representation of the model, and put into the MD of knowledge base in accordance with the users' desire, after they selected model type in the process of model layout. MG structure is designed and shown in Fig. 3.

In visual programming language, SPATIAL PARSING is one of the most important techniques. If the image can be viewed as a visual communication object which can be interpreted, spatial parsing is the first step in interpretation. In fact, spatial parsing is the process of recovering the underlying syntactic structure of a visual communication object from its spatial arrangement. A good survey on spatial parsing technique has been presented by Fred Lakin[7].

In the design of the development tool, we borrow the idea of spatial parsing of visual language. However, there is a little difference between them. In our system, spatial parsing can be defined as the process of recovering the spatial arrangement, namely, model layout. During making model layout, a coordinate system was erected in the working region. For a component, there is always a minimum enclosing rectangle(MER) uniquely associated with the component on 2D plane, mathematically written as MER(Xc,Yc,Xmin,Ymin,Xmax,Ymax), where Xc,Yc are coordinates of region controid. During spatial parsing,the first step is to obtain every MER components of a model layout. Because of the coordinate system existed, After MER is obtained, on the assumption that every MER is only point objects, we can easily recover simple relations such as north, east,west and south as well as their aliases-- over, under, and next to-- by means of some simple calculations. Besides the simple spatial relations n question, we need to recover more complicated and necessary spatial relations such as surrounded by, partly surrounded by,same as, and so on. we assump that one of the components is a point of view component(PVC) and then views the other component in four directions(north, east, south, west). Hence, at least one or at most four subparts of the other component can be seen from the PVC. Once a PVC has been selected, we can represent complicated spatial relation using simple spatial relations such as east, south, west and north. Some other spatial relations, such as north-ease, north-west, south-east, south west are also easily obtained by simple reasoning. For example, north-east(X) is north(X) and east(X), where X is the label of a component. Using the approach, a complicated component can be a labeled, segmented image. Each connected part has a unique label. If the component with a label X is a PVC, a component with the primary label Y is segmented into Y_1, Y_2, ..., Y_m from different directions based on X, and Y_i and Y_j $(1 \leq i,j \leq m, i != j)$ are located in the west and the north, respectively. we know that the component with the label Y is connected; therefore we can at least deduce that X is partly surrounded by Y on its west and north sides. In a similar manner, we can find the surrounded by relations by looking for combinations of those patterns which give rise to this situation. We must have a component with one part south, east, north and west of a component with (at least) four parts. In deriving the surrounded by and partly surrounded by relations, a model with a large number of components will generate a large number of partial solutions as the algorithm attempts to find the north, south, east and west relation as subgoals and then unify those subgoals at higher level. Many of the supurious partial solutions can be suppressed by recalling that an enclosing component must have multiple parts.

A recursive application of the above principle can lead to a desired representation. It is important for the representation to translate from the model layout (visual form) to the model symbolic representation. Although the development tool is designed to support a special purpose image understanding model, to some extent, general purpose is also taken into account as one of our designing criteria. In order to generate multi-representation forms, a proper medium form is needed. In this system, quadtree has been selected employed as the medium form. After a spatial parsing, the quadtree is easily generated. Usually, the root node of the quadtree is the compnent with a key tag. Its four branches reflect the situation of its spatial relation with other components of the model, viz., north, east, south and west when it is viewed as PVC. Except the root node, other nodes are only one of the following types: component, a part of component and null node. All son nodes of a node denoted N in quadtree, if it has, are the results of a spatial parsing when node N is viewed as PVC.

After a quadtree has been generated, the translation from the medium form into a special symbolic representation can be completed by a traversal of the quadtree and a special algorithm. Up to now, however, we have only developed a translation algorithm from quadtree to semantic nets. We think it is not difficult to find the rest translation algorithms. Hence, the detail discussion on them is not presented in the paper.

3.4 System Maintenance

Considering the integrity of the system, we present a set of commands to aid users to maintain the system. These commands include the following types: SYMBOLIC MODEL STORAGE, MODEL RETRIEVE, MODEL DELETION&MODIFICATION and CONSISTENCE CHECK.

Symbolic model storage consists of two commands, viz., store and restore. After a model is generated, ussally, it needs to be put into MD. Store command serves this objective. Restore command is used when a model existed in MD needs to be modified. After the model has been modified, restore command can put this new model into MD. This kind of commands is implemented by means of Unix file system.

Model retrieve is indispensable to the system. First of all, it can supply an efficient tool for users to check MD. The next is that it is the first step of modifying a model in MD yet. Because there are MERs left after a spatial parsing, model layout (visual form) can be obtained easily from their symbolic representation.

Model Deletion&Modification is an essential funtion of system maintenance. In fact, this funtion is a necessary for the management of any knowledge base. Model deletion is simply implemented by cancelling the model and its attributes from MD and releasing the storage space occupied by them. Model modification can be implemented through a complicated process as follows: the first is retrieval, the following is modification of the original model layout with an interactive input device, and the final is generating a new symbolic form of the model and restoring the model to MD.

Consistence check also plays a particular role in system maintenance. It includes two cases, viz., same name with different model layout and different name with same model layout. Consistence check works when it happens that a vew model is generated or a model existed in MD is modified.

Once either of two cases are checked, an application of rename technique can keep the consistenance of the system.

4. Discussion

In the section, we discuss two topics in future research and a performance measures for this system. The first topic is why and how to aid a knowledge base to acquire visual knowledge. The other is some issues on general purpose spatial parsing in visual programming language.

As mentioned in the introduction, visual knowledge is a common type of knowledge. The following premises can give a strong evidence to support the significance of visual knowledge acquirement. (1) People, in general, prefer pictures over words. (2) Pictures are more powerful than words as a means of communication. They can convey more meaning in a more concise unit of expression. (3) Pictures do not have the language barriers that natural languages have. They are understood by people regardless of what language they speak. Hence, it should be highlighted to aid a knowledge base to acquire visual knowledge. This research has shown that it is necessary and efficient for a knowledge base related to visual knowledge to bulid up a software tool to make users input visual knowledge directly. Although the generalized icons theory and visual programming language techniques have been developed in recent years, it is our future research topics to improve their techniques and to extend applications of them.

Up to nows. however, there have been no general purpose spatial parsing One of our resent research topics is the general purpose spatial parsing research by supplying an efficient specification language to manage users input icon data and developing a software tool to aid spatial parsing. In addition, we are highly interested in research topics on the medium result representation after a spatial parsing.

N.C. Shu proposed a dimensional framework for the analysis of visual programming language so that they can be characterized and compared[8]. According to the creteria, we think that our system is high in visual content, low in language level and middle in scope.

5. Conclusions

We are currently implementing the development tool using the C programming language on SUN 3/260 running Unix operating system and configuring with GKS graphic software. From the viewpoint of engineering, this research reflects the following design philosophy that some development tools ought to be developed in advance to support to develop a large scale complex system. Usually, these development tools can aid user to efficiently and easily develop huge system and the results obtained by these development tools are high qualities, even better than ones developed directly by people.

References

[1] D.H. Ballard and C.M. Brown, Computer Vision, Prentice-Hall Inc., 1982.

[2] Ke Chen and Z.R. Li,"Image Understanding Based on RBC Theory," Proceedings of The Third Pan Pacific Computer Conference, Beijing, China, International Academic Publishers, August 1989, pp. 906-911.

[3] Irving Biederman,"Recognition-By-Components: A Theory of Human Image Understanding," Psychological Review, Vol. 94, No.2, 1987, pp. 115-147.

[4] Ke Chen, Y. Cheng and Z.R. Li," A Model-Directed Image Understanding System for Computer Vision," submitted to IEA/AIE-90, Charleston, SC, U.S.A., July 1990.

[5] Shi-Kuo Cheng, Principles of Pictorial Infomation System Design, Prentice-Hall International Inc., 1989, pp. 303-321.

[6] Shi-Kuo Cheng,"Visual Languages: A Tutorial and Survey," IEEE Software, January 1987, pp. 29-39.

[7] Fred Lakin,"Spatial Parsing For Visual Languages," in Visual Languages, eds. S.K. Cheng et al., Plenum Press, 1986, pp. 35-86.

[8] N.C. Shu,"Visual Programming Languages: A Perspective and A Dimensional Analysis," in Visual Languages, eds. S.K. Cheng et al., Plenum Press, 1986, pp. 11-34.

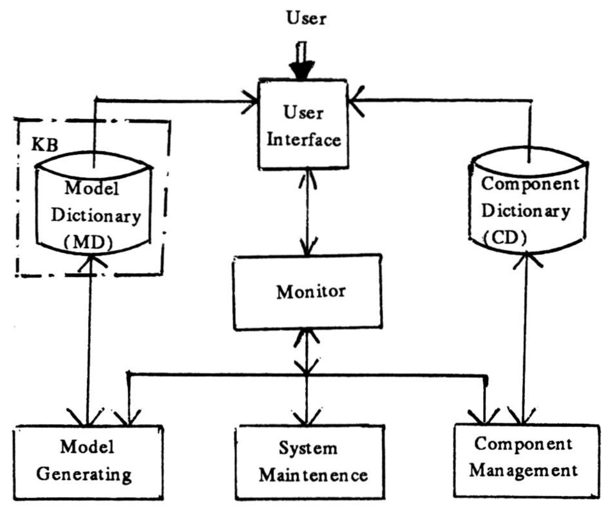


Fig 1. The development tool architecture.

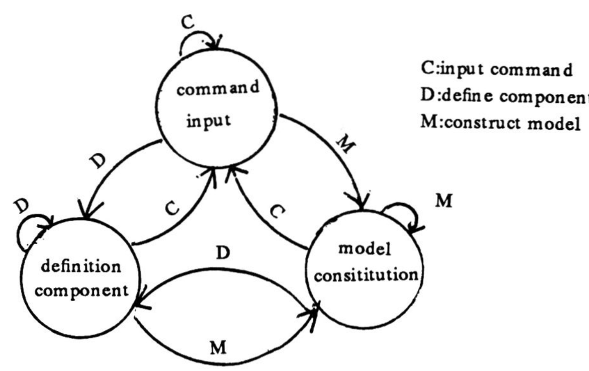


Fig 2. A state transition diagram.

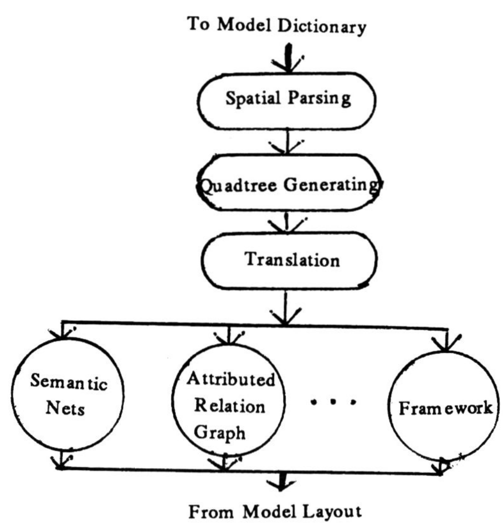


Fig 3. Model generating module structure.