MVA'90 IAPR Workshop on Machine Vision Applications Nov. 28-30,1990, Tokyo

Descriptive Ability of Drawing Image Understanding Framework Using State Transition Models

Shin'ichi SATOH, Masao SAKAUCHI

Industrial Institute of Science, The University of Tokyo
7-22-1, Roppongi, Minato-ku, Tokyo 106, Japan

Abstract

It is strongly desired to construct multimedia databases especially of maps or engineering drawings. If these databases are intended to be used really effectively, original drawing images should be recognized as completely as possible. We have been proposed a new framework of the drawing image understanding system to achieve this. This framework has following features: a) the understanding knowledge is free from the understanding mechanism, b) the understanding knowledge can be obtained relatively easily, c) the understanding mechanism can be used commonly among its applications.

Some understanding systems based on this framework have been implemented, and it is found that some kind of drawings can be applied this framework very easily, but some of them are not so easy to apply. In this paper, outline of this framework is described first. Then applying this framework to some kinds of drawing images, we reveal that what kind of drawings are applicable and what kind of them are not. And discussion of how to extend our framework to apply this to the latter drawings is presented.

1 Introduction

As demand for constructing an effective multimedia database of drawings such as maps or engineering drawings increases in recent years, more powerful and flexible drawing image understanding system is strongly desired. There have been many researches for drawing understanding systems including applying production system[1,2,3], hypothesis verification method[4], truth maintenance system[5], and so on. They, however, are still unsatisfactory. Some of the reasons can be considered as following:

- These systems are often strictly related to its own application domain, so the system to recognize some kind of drawings cannot work for other kind of drawings. To apply such system to different kind of objects, the system should be fully reconstructed.
- Image understanding technology itself is still immature to get all necessary information from images to construct a practical multimedia database.

Because the multimedia database for drawings is often expected to cover very large and various types of data, above problems should be overcome.

We have been creating a new paradigm[6] to be a general framework of drawing image understanding system intending to resolve above points. This framework has the following features.

- The understanding system is constructed with two parts. They are an understanding system kernel that is fully independent of its application domain, and understanding methods that is deeply dependent on its application domain.
- Understanding methods are given as abstraction rules independently to a understanding system kernel.
- The understanding kernel is independent of its application domain, so it can be controlled flexibly.

Because the understanding kernel is independent of its application, it can be used commonly for various kind of drawings while only understanding rule sets vary from drawings to drawings. These rules are given rather declarative than procedural, so they can be obtained relatively easily.

Though understanding rules can be described relatively easily, they should be described with great care to suit the understanding system to a new application. We have evaluated this framework as drawing image understanding systems for some different kinds of drawings, and found that it was quite easy to construct rules for some drawings but it was not so easy for others. It is necessary to reveal the relation between the descriptive ability of these rules and the property of drawings.

In the followings, first the outline of this framework is presented, and then some evaluations of its descriptive ability of these rules is shown with some example understanding systems.

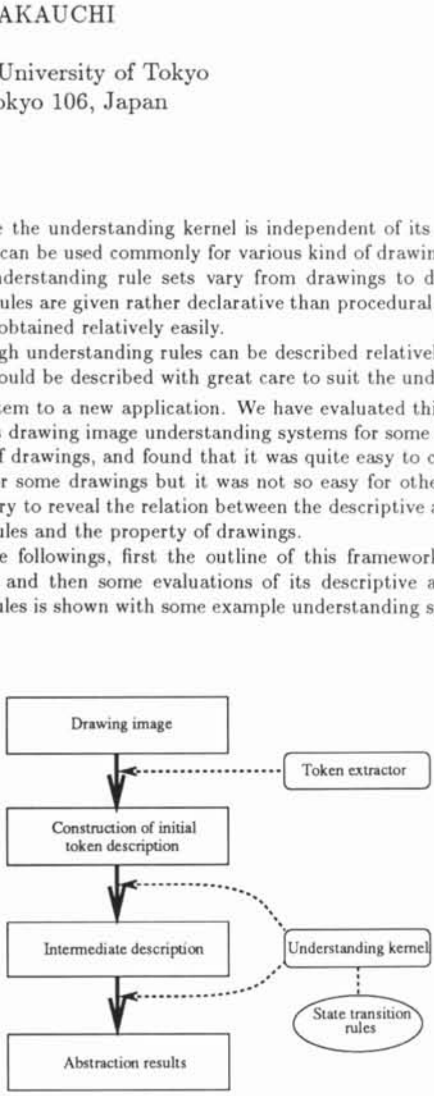


Figure 1: Framework of the model

2 Outline of the framework

Fig. 1 shows the understanding system structure based on our framework. Drawing images are typically given as bitmap images processed by the scanner. First, this image is converted to an initial token description by a token extractor. An token description of drawing gives symbolic representation of an image and it is composed of a variety of tokens, which have geometrical entities such as edges, lines, bars, boxes, dots, and so on.

Next this description is processed by the understanding kernel (fig. 1). In our framework, the understanding process is modeled to make each token correspond to an appropriate abstraction label. The understanding kernel performs this labeling process. These processes are realized by the understanding kernel as transition of each token's internal state. Each token has its own internal state. Tokens inspect surrounding circumstances with each

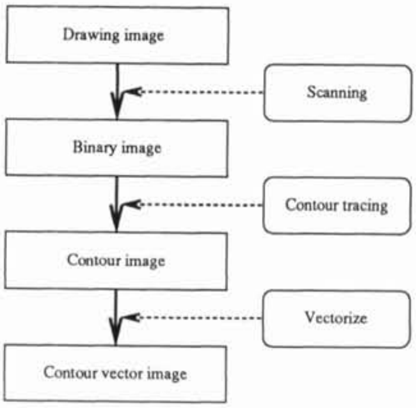


Figure 2: Outline of the drawing processor AI-Mudams

other, and they change their state independently and successively. These series of state transitions correspond to the understanding process of drawing images itself.

2.1 Token extractor

A drawing image are usually given as a bitmap image. This bitmap image is 2 dimensional array whose elements represent intensity at corresponding place of the image. The token extractor generates a symbol level description from a given bitmap image for an following understanding process. We define these geometric symbols as tokens.

Though there are many image processing methods suitable for token extracting, we use the drawing image processor AI-Mudams[7,8] for the embodiment of the token extractor. The AI-Mudams, whose outline is shown in fig. 2, is a high-speed software-based drawing image processor developed by authors' group. In the AI-Mudams, bitmap image data are first converted to suitable pattern primitives with much higher abstraction label than image pixels, such as contour vectors, segments, by an appropriate image data processing.

Using the AI-Mudams, a token description is basically given by the form of series of line segments, which correspond to an core line of contours of objects in the bitmap image, of which crossing part detached. The description produces by AI-Mudams has many features suitable for an initial description of an understanding process. For example,

- An image is fully described with the symbols of line segment.
- The information of the given bitmap image can also be referred almost completely when needed.
- The physical size of the information is relatively small.

For practical reason, the actual token extractor classifies contour segments into a group representing comparatively long lines and others representing comparatively small symbols. And then the extractor convert the line group into line tokens, and the symbol group into symbol tokens.

2.2 Understanding kernel

The understanding kernel is the core part of this understanding system. It provides understanding process for the given initial token description. This understanding process is realized to label each token as its appropriate interpretation label.

In our model, each token is defined as an individual active agent which has its own internal state, can inspect surrounding circumstances geometrically, and has its own knowledge to determine its next state corresponding to its last state and its circumstances. Tokens interact in parallel, so recognition proceeds step by step everywhere.

According to this model, an understanding system kernel shown in fig. 1 just simulates each token as an individual agent interacting together and changing its state. In addition, understanding knowledge is given as state transition rules of each agent, so specific description of an application domain (which corresponds to an individual types of drawing) exists only in state transition rules. This means that this understanding system kernel is free from its application domain, and can be used in common among understanding systems for different kinds of drawings. If the need for the understanding system of a new kind of drawing arises, the only thing required is to reconstruct the state transition rules suitable for the new problem. The system kernel has responsibility for everything not essential to state transition knowledge such as transition scheduling and the scenario of understanding process. As the result, transition rules can be free from an understanding mechanism itself.

The understanding kernel works in two modes; one is a bottom-up process mode, and the other is a top-down process mode. In the bottom-up process mode, each token transits its state determinately and reduces alternate choices of labeling to speed up the understanding process in top-down mode. In the top-down process mode, state transition process can backtrack to any choice point, so nondeterminate recognition can be realized.

In our embodiment, the token extractor AI-Mudams written in C language, and the understanding kernel written in Prolog and C are implemented. Especially the understanding kernel is mainly written in Prolog, and the state transition rules are described in the form of Prolog.

3 Evaluation of our framework

In this section, evaluation of our framework with some typical example drawings is shown. Map drawings and mechanical drawings are given as typical examples. The understanding process is realized as series of state transitions of each token along relations between tokens such as is-a or is-part-of, so it is necessary to reveal these relations, that is a state transition diagram, to evaluate behavior of the understanding system. According to this relations, it is examined how to describe these state transition rules. After these considerations, it is found that there are some kinds of drawings which it is so hard to compose the rules for. And last some countermeasure to improve the descriptive ability is shown.

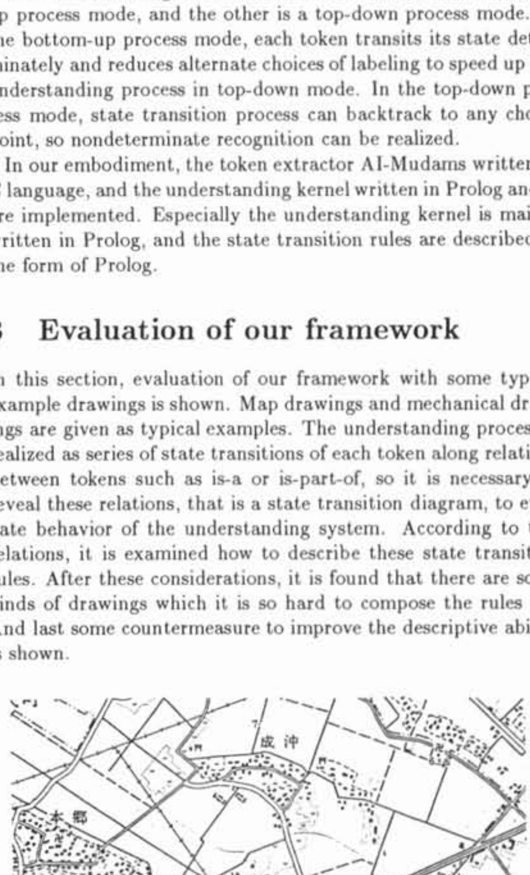


Figure 3: A typical map drawing

3.1 Map drawings

Here we assume to apply our framework to the map drawings understanding system. Fig. 3 shows a typical map drawing familiar in Japan. Our goal is to extract some semantic components from such kinds of maps. These components are following.

roads Roads are expressed as long stroke solid lines. And there are another expression of roads expressed as long stroke dashed lines, which are relatively narrow roads. So they are obtained as series of long bar symbols.

land usage boundaries Land is labeled according to how it is used such as farmland, forest, and so on. Land usage boundaries are boundaries of land which distinguish its usage. So they are expected to be closed loops on maps. These boundaries are formed mainly with dotted lines — series of dots — and can include some part of roads and special area boundaries which is explained below. Land usage boundary surrounds some symbols which show the usage of this boundary.

special area boundaries Special area boundaries are boundaries of some special area like residential section etc. They are also expected to be closed loops. These are formed mainly with short stroke dashed lines, which are series of short bar symbols, and with roads. Special area boundary usually surrounds some house symbols.

some other symbols The map drawings contain some kinds of symbols other than symbols which form above components. They represent houses, land usage symbols, and other solitary symbols.

Fig. 4 shows a relation diagram between tokens that form map drawings which is constructed with careful inspection of features of above components. Tokens transit their state along these relations. Referring to this diagram, it is found that there are not many nodes in the diagram, because map drawing is composed of not so many components. It is not so easy to recognize these components simply because there are not many kinds of components, as there are many complicated constraints to be satisfied to transit to a state from some states, for example, to transit to the state land-usage-boundary from series-of-dots, road, and special-area-boundary (they are related to land-usage-boundary with relation of is-part-of as shown in fig. 4).

But it is relatively easy to compose understanding rules for this kind of drawings that are composed of not so many tokens like maps. Actually, the understanding system that recognizes above components of maps has been implemented and tested, for the map image processed by AI-Mudams shown in fig. 5 a) is successfully processed into the image shown in fig. 5 b), although this system should be loaded 83 complicated rules.

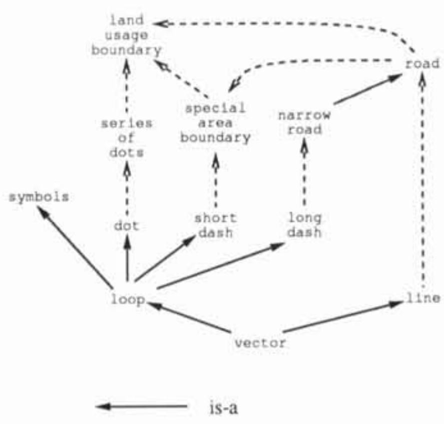
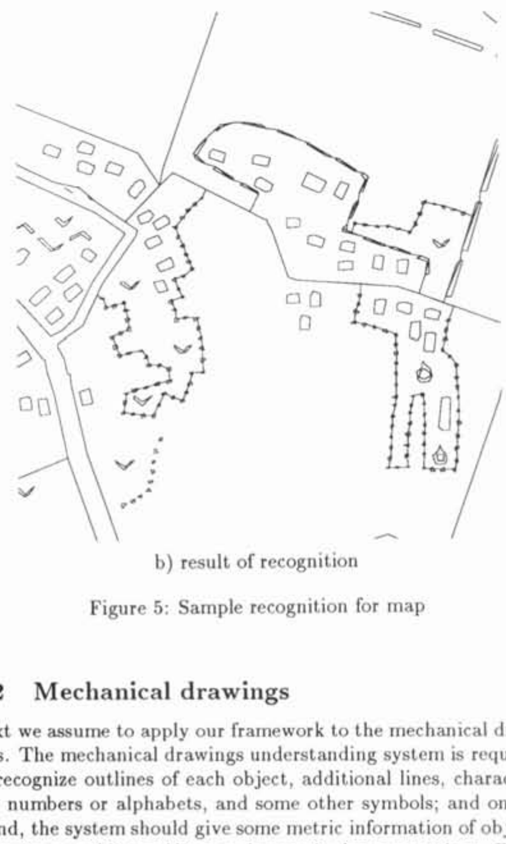


Figure 4: A relation diagram between tokens for recognition of map

a) token description



b) result of recognition

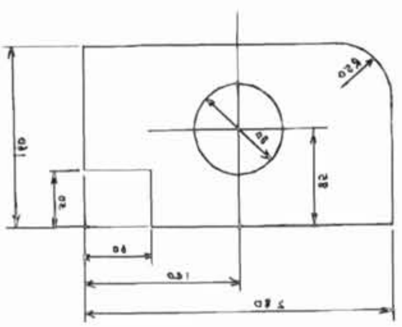
Figure 5: Sample recognition for map

3.2 Mechanical drawings

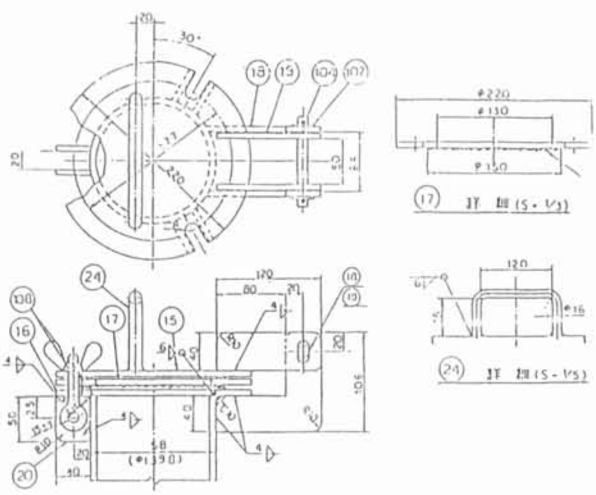
Next we assume to apply our framework to the mechanical drawings. The mechanical drawings understanding system is required to recognize outlines of each object, additional lines, characters like numbers or alphabets, and some other symbols; and on demand, the system should give some metric information of objects according to additional lines and numeric character strings. Fig. 6 shows samples of mechanical drawings, where fig. 6 a) is a very simple one. This sample consists of only a sheet type object and some additional lines and character strings. So our understanding system can recognize them properly like a map drawing example above.

Fig. 6 b) is a very complicated one, and as a practical mechanical drawing understanding system a mechanical drawing understanding system is expected to understand such drawings. Nevertheless, it is difficult to apply our framework as it is to such drawings. Some of difficulties to understand these drawings are considered as following.

- These drawings are composed of so many parts like bolts, nuts, shafts, pipes and various shapes of sheet metals, so the



a) simple example



b) complicated example

Figure 6: Examples of mechanical drawing

understanding system should have models of these parts as rules.

- Occasionally configurations of parts are also given as drawings themselves. A target drawing should be recognized according to its parts configurations given as other drawings.

To apply this framework to such kind of drawings, it is necessary to load huge number of rules which correspond to models of parts into the understanding system. It is not practical to implement all these rules for parts which may be included in the target drawing or may not. Besides our framework does not have any mechanism to give some configurations of parts as other drawings.

3.3 Considerations

Above subsections reveal that some types of drawings are recognized with our framework relatively easily, but other types of them are not. The former drawings consist of only a limited number of components, although some of them have very complicated configuration. Understanding rules for these drawing can be obtained relatively easily, and the understanding system based on our framework can recognize these drawings successfully. These include some kind of map, logic circuit diagram, flowchart and so on. The latter drawings consist of a lot of components, and in addition some configurations of components are given as other drawings. These drawings include design drawings, mechanical drawings and so on.

Our framework can not deal with these drawings as it is. To process these drawings, our framework should be extended in some way. To make it applicable to drawings that include infinite numbers of components the understanding system should refer to external databases. These databases may be based on the semantic network data model, or object oriented data model. And next, to understand target drawings according to configurations given as other drawings, the understanding system should recognize these configuration drawings first, and update its internal understanding knowledges according to results of this recognition. This may be based on the notion of the reflection or the computer learning.

4 Conclusion

A new drawing image understanding framework based on state transition models is introduced, and some considerations of applicability of this framework are given. To sum them up, drawings which have not many components and models of all components of them can be described as rules can be recognized successfully with this framework. But to apply this framework to drawings which have so many parts, it is necessary to make the understanding system refer to external databases. And to apply it to drawings which are given their parts' configurations as drawings, the understanding system should update its knowledge by itself according to these configurations. These points are now examined.

References

[1] D. M. McKeown. The role of artificial intelligence in the integration of remotely senced data with geographic information systems. *IEEE Trans. on Geoscience and Remote Sensing*, GE-25(3):330–348, May 1987.

[2] D. M. McKeown. Rule-based interpretation of aerial imagery. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-7(5):570–585, Sep. 1985.

[3] F. Wang et al. A knowledge-based system for highway extraction. *IEEE Trans. on Geoscience and Remote Sensing*, GE-25(3):349–359, May 1987.

[4] T. Matsuyama. Expert system for image processing – knowledge-based composition of image analysis processes – . In *9th Inernational Conference on Pattern Recognition*, pages 125–133, IEEE, 1988.

[5] J. de Kleer. An assumption-based TMS. *Artificial Intelligence*, 28:127–162, 1986.

[6] Shin'ichi Satoh, Yutaka Ohsawa, and Masao Sakauchi. Drawing image understanding framework using state transition models. In *10th ICPR*, pages 491–495, IEEE, 1990.

[7] Wei Lu, Yutaka Ohsawa, and Masao Sakauchi. A database capture system for mechanical drawings using an efficient multi-dimensional graphical data structure. In *9th ICPR*, pages 266–269, IEEE, 1988.

[8] M. Sakauchi. Two interfaces in image database systems. In *Proc. of The International Workshop on Industrial Applications of Machine Intelligence and Vision*, pages 22–27, IEEE, 1989.