MVA'94 IAPR Workshop on Machine Vision Applications Dec. 13-15, 1994, Kawasaki

# A METHOD OF UNDERSTANDING CONCEPTUAL DIAGRAMS *

Noriyoshi Yoneda Koichi Kise Shinobu Takamatsu Kunio Fukunaga

Department of Computer and Systems Sciences
University of Osaka Prefecture
E-mail: (noriyosi | kise)@ss.cs.osakafu-u.ac.jp

## Abstract

*A conceptual diagram is a line drawing which represents semantic structure of concepts using simple geometric entities. This paper presents a method of understanding conceptual diagrams. The objective of our method is to interpret semantic roles of geometric entities in conceptual diagrams. In conceptual diagrams, however, a single geometric entity plays various semantic roles for representing concepts, because there are no strict rules for writing conceptual diagrams. To cope with this problem, we introduce the strategy of hypothesis generation and verification; hypothesized interpretations are verified by relaxation which takes account of the semantic relation to other entities. From the experimental results using 50 conceptual diagrams, we discuss the effectiveness and the limitations of our method.*

## 1 Introduction

Understanding of line drawings is indispensable to realize document image understanding. A number of studies have been made on various types of line drawings (e.g., technical drawings, maps, flow-charts and circuit diagrams). In the interpretation of line drawings, most of the existing methods focus on extracting precise description of geometric entities and their spatial relations (or physical structure). It would be sufficient for understanding flow-charts and circuit diagrams, because, by the rules of writing these diagrams, the physical structure clearly corresponds to what these diagrams semantically represent; once the physical structure is reconstructed, it is trivial to extract the information represented in these diagrams.

In recent years, however, the need to extract semantic entities and their relations (or logical structure) has been emphasized[1, 2]. It seems essential for some kinds of line drawings like *conceptual diagrams*. As shown in Fig. 1, a conceptual diagram is a line drawing which illustrates the structure of some concepts using simple geometric entities (loops, lines and character strings). Conceptual diagrams are similar to flow-charts in physical structure. However, the logical structure should be interpreted from the physical structure, because there are no rules which make the interpretation trivial.

In this paper, we propose a method of understanding conceptual diagrams. A major problem in understanding conceptual diagrams is that a single geometric entity plays various semantic roles depending on surrounding entities. For example, a line in Fig. 1 can be interpreted as a relation between concepts (solid lines), division of concepts (dashed lines [1]); geometric entities cannot be unambiguously interpreted from local viewpoints. To cope with this problem, we introduce the strategy of hypothesis generation and verification. Hypothesized interpretations of physical structure are verified by relaxation which takes account of the global consistency of logical structure. From the experimental results using 50 conceptual diagrams, we discuss the effectiveness and the limitations of our method.

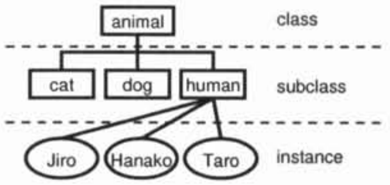


Figure 1: An example of conceptual diagrams

## 2 Conceptual Diagrams

Let us start with considering the physical and logical structure of conceptual diagrams. The physical structure can be represented as physical relations between physical objects as follows:

**physical objects** loops (rectangles, ovals, etc.), lines (solid, broken and dotted lines with or without arrowheads) and character strings (simply called strings hereafter).

**physical relations** spatial relations between physical objects such as contact, overlap, proximity and alignment.

The logical structure is likewise represented as logical relations among logical objects.

**logical objects** concepts represented in a diagram. Concepts often have their labels represented as strings.

**logical relations** relations among concepts. Although there would be many kinds of relations among concepts, we focus here on the relations explicitly represented in a diagram. Labels are often attached to logical relations.

It can be generally said that a person who writes a line drawing determines its physical structure aiming at easy understanding of its logical structure. In other words, the physical structure of a line drawing is closely

*This work was supported in part by Grant-in-Aid for Scientific Research from the Ministry of Education, and the Telecommunications Advancement Foundation.

[1] Dashed lines are used just for explanation. If they were written in solid lines, they would still represent division of concepts.

related to its logical structure. For flow-charts and circuit diagrams, such relation is strictly determined as standards. However, there are no standards or definite rules for conceptual diagrams; we only have some customary rules of writing conceptual diagrams.

The difficulty of understanding conceptual diagram is attributable to this point. To be concrete, we face various local ambiguities in interpretation of physical structure. Some of them are listed below:

- Concepts are often represented as loops. However, there exist concepts represented in different ways. For example, a string can solely correspond to a concept. Similarly, a compound concept is often represented as a loop which encloses some loops. However, aligned loops sometimes (but not always) indicate the existence of a compound concept including concepts represented as the loops.
- Lines often corresponds to logical relations among concepts. However, lines also represent the division and the grouping of concepts.
- Strings are often interpreted as labels of concepts/ relations which are represented as loops/lines. However, it is not easy to find which loop/line a string is associated with; a string is not always the label of the loop/line closest to the string.

## 3 Overview of Processing

Our method of understanding conceptual diagrams is twofold: extraction of physical structure and extraction of logical structure.

The process of extraction of physical structure takes as input the data of line segments and strings, and generates the description of physical structure. In the input data, a line segment is represented as coordinates of two end points, a type of a line (solid, dotted or broken) and a type of each end (with or without an arrowhead). A string is represented as coordinates of its bounding rectangle and characters in it.

The description of physical structure is interpreted by the process of extraction of logical structure. To cope with the local ambiguities, we employ the strategy of hypothesis generation and verification. First, from local viewpoints, possible interpretations of the description are enumerated as *hypotheses* of concepts, relations and labels. Then, these hypotheses are verified to reject unplausible interpretations.

Note that we do not deal with the linguistic meaning of concepts; we aim to extract the logical structure explicitly represented in a diagram. Thus logical objects and relations which have no labels are accepted as the output, and no further processing such as identification of the hidden meaning of logical objects or relations[2] is considered.

## 4 Extraction of Physical Structure

This process consists of the extraction of physical objects, physical relations and implicit loops indicated by physical structure.

### 4.1 Physical objects

As described in **2**, loops are important physical objects in conceptual diagrams. Thus we attempt to extract loops from line segments. By extracting all loops from line segments, we can also obtain lines from the rest of line segments. Our procedure resembles the one described in [3], except that no explicit models of loops are utilized.

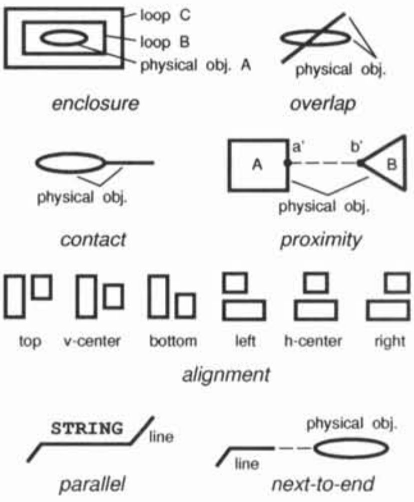


Figure 2: Physical relations

Ends of line segments are classified into terminals, paths and branches. A *terminal* end is the end which belongs to only one line segment. A *path* end is the end at which exactly two line segments contact. A *branch* end is the end at which more than two line segments contact. A *chain* is a sequence of line segments concatenated at all path ends on condition that: (1) a chain includes line segments of the same type (solid, dotted or broken), (2) a chain does not include an arrowhead in the middle.

In the first step, all chains are extracted from the input data of line segments. Next step is to find apparent loops and lines. A chain whose two end points coincide is identified as a loop and removed from the input data. If a chain has at least one terminal end, or has at least one arrowhead, it is identified as a line and removed. This step of processing is repeated until no more chains are removed. In the third step, we focus on chains connected at a branch end. If two of such chains form a straight segment at the branch, they are concatenated. Then loops are extracted again from chains. After all loops are extracted, the chains which remain in the input data are regarded as lines.

### 4.2 Physical relations

As the physical relations, we consider the relations shown in Fig. 2. In the followings, the bounding rectangle of a string is considered, in the case that a physical object is a string.

The relation *enclosure* is defined between a loop and a physical object. If a loop includes a physical object and no other loops do not include both the physical object and the loop, it is said that the loop encloses the physical object, or the loop has the enclosure relation to the physical object. In Fig. 2, the physical object $A$ is enclosed by the loop $B$, but not by the loop $C$. Two physical objects *overlap* if one of the physical objects lies inside the region bounded by the other physical object. *Contact* is the relation between two physical objects if their boundaries share some points and they do not overlap. For the relation *proximity*, we focus on the distance between physical objects $A$ and $B$. The

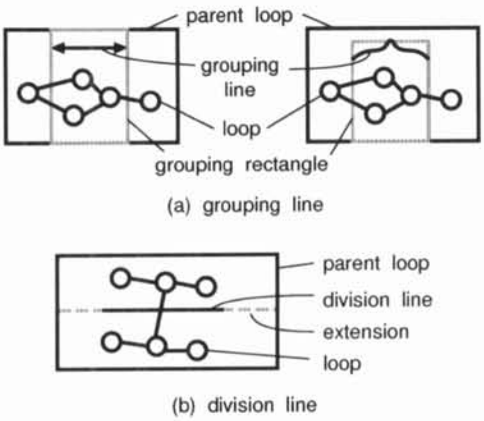


Figure 3: Grouping and division lines

distance is defined as the minimum distance $d_{min}(> 0)$ between points $a$ and $b$ which are on the boundaries of $A$ and $B$, respectively. The points forming the minimum distance are described as $a'$ and $b'$. If $d_{min}$ is less than a certain threshold and no physical objects overlap with the segment between $a'$ and $b'$, $A$ has the proximity relation to $B$. For the *alignment* relation, we utilize six types shown in Fig. 2.

The relations parallel and next-to-end are somewhat special. The relation *parallel* is defined between a line and a string. If a line segment in a line is parallel with the longer side of the bounding rectangle of a string, they have the parallel relation. The relation *next-to-end* is the special case of the proximity and the contact. If (1) a line has the proximity relation to a physical object, and (2) the extension of the line from an end contacts with the physical object, the line has the relation of next-to-end with the physical object. In addition, if an end of a line contacts with a physical object, they also have the next-to-end relation.

## 4.3 Implicit loops

The role of this step is to identify a group of loops represented by grouping and division lines, and alignment of the loops. Examples of grouping and division lines are illustrated in Fig. 3, where the parent loop indicates either a loop or a bounding rectangle of a diagram.

As shown in Fig. 3(a), a grouping line is the line which satisfies the following conditions:

1. Both of the two ends of the line have arrowheads, or both of them have no arrowheads.
2. The shape of the line is straight, or like a brace.
3. The line must not overlap with the loops in the parent loop.
4. The grouping rectangle shown in Fig. 3(a) encloses some but not all loops in the parent loop, and none of the loops in the parent loop overlaps with the boundary of the grouping rectangle.

On the other hand, the conditions of a division line are as follows (see Fig. 3(b)):

1. The line does not have an arrowhead.
2. The line does not overlap with the loops in the parent loop. If the line does not contact with the parent loop, we also consider the extension shown as the broken line in Fig 3(b).

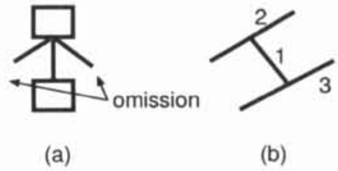


Figure 4: Logical relations

3. The line divides the loops in the parent loop into at least two groups. We consider the extension of the line similar to the above condition. A bounding rectangle of a group of loops is also called a grouping rectangle.

In the case that these two types of lines or the loops having the alignment relation are identified, a group of loops is extracted as an *implicit* loop which is represented as a grouping rectangle. When an implicit loop is identified, physical relations about the implicit loop are also calculated. In the following, we use the term *explicit* loops to refer to loops except implicit loops.

# 5 Extraction of Logical Structure

## 5.1 Hypothesis Generation

In this step, all possible interpretations of physical objects and relations are enumerated as hypotheses from local viewpoints. Hypotheses generated at this step are classified into three types: hypotheses of logical objects (concepts), logical relations (relations among concepts) and labels (names of concepts or relations).

Hypotheses of logical objects are generated from the following physical objects:

- loops (explicit and implicit),
- strings having the relation next-to-end to lines,
- dotted or broken straight lines. (These lines indicate the omission of logical objects.)

Hypotheses of logical relations are generated between physical objects as follows:

- a line having the relations of next-to-end to physical objects (a logical relation between the physical objects).
- the enclosure relation between loops (a logical relation "part-of" between the loops).

Note that a line which has the next-to-end relation at only one end is also accepted as a hypothesis of a logical relation, because a physical object is sometimes omitted as shown in Fig. 4(a). In such cases, we also generate a hypothesis of an omitted logical object. In addition, a line is interpreted as a logical relation with other lines. In Fig. 4(b), the line 1 is hypothesized as the logical relation between the lines 2 and 3. This enables us to interpret a set of lines as an n-ary relation.

Hypothesis of labels are generated for each string accompanied with a physical object with which the label is associated. A simple way to do this is to associate a string with physical objects each of which has the proximity relation to the string. However, this may cause many incorrect hypotheses or miss many correct hypotheses depending severely on the threshold of the proximity relation. Thus, we utilize some heuristics to improve the accuracy of hypothesis generation. Hypotheses of labels are generated as follows:

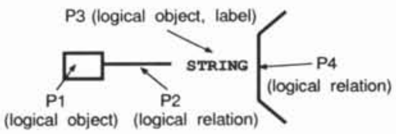


Figure 5: An example of hypothesis verification

- a string is hypothesized as the label of a loop which encloses the string.
- (heuristic 1) If a loop has the relation of alignment to a string in addition to proximity, the string is hypothesized as the label of the loop.
- (heuristic 2) If a line has the relation of parallel to a string in addition to the proximity, the string is hypothesized as the label of the line.
- If a string does not satisfy both of the above two heuristics, the string is hypothesized as the labels of loops and lines which have at least one of the proximity, contact and overlap relations with the string.

### 5.2 Hypothesis Verification

The following constraints are employed to verify hypotheses.

**C1** A physical object except implicit loops must be interpreted as at least one of a logical object, a logical relation and a label.

**C2** A logical object must have a logical relation.

**C3** A logical relation must have two logical objects to be related.

**C4** A label must have a logical object or a relation to be associated with.

**C5** An implicit loop except ones generated from division lines must have a label.

**C6** A dotted or broken line which represents the omission of logical objects must not have a label.

The procedure of hypothesis verification behaves like *relaxation*. Rejection of invalid hypotheses found by testing **C2**–**C6** is repeated until no more hypotheses are rejected. The verification fails if **C1** is violated by the rejection.

Let us consider a simple example shown in Fig. 5. Hypotheses for physical objects P1–P4 are listed in parentheses. The physical object P3 has two interpretations (a logical object and a label of P4), while other physical objects have only one interpretation. We can select the interpretation "P3 corresponds to a logical object", since **C3** is violated if P3 is a label of P4.

### 5.3 Selection of plausible hypotheses

The constraints utilized in the verification are not strong enough to select the most plausible hypotheses. In particular, incorrect hypotheses of labels remain after the verification. In order to select the hypotheses of labels, we utilize the following rules: (1) If a string overlaps or contacts with a physical object, a hypothesis stating that the string is attached to the physical object is selected. (2) Otherwise, a hypothesis stating that a string is attached to the nearest ($d_{min}$ is smallest) physical object is selected.

Table 1: Experimental results

| | No. of hypotheses($N$) | Cover rate ($C$) |
|---|---|---|
| generation | 1.53 | 99.7% |
| verification | 1.16 | 99.7% |
| selection | 1.00 | 99.2% |

## 6 Experimental Results

Our method was applied to 50 samples of conceptual diagrams obtained from various technical papers and textbooks written in Japanese and English. In these samples, 471 logical objects, 517 logical relations and 491 labels were included.

Results were evaluated at each steps of extraction of logical structure (i.e., hypothesis generation, verification and selection) using the following criteria:

$N$: the average number of hypotheses for one correct logical entity (i.e., an object, a relation or a label),

$C$: cover rate: the rate of the number of correct hypotheses for the number of correct logical entities.

Table 1 shows the experimental results. At the step of hypothesis generation, two correct logical objects could not be hypothesized since labels were too apart from their physical objects. In addition, lines crossing perpendicularly as in Fig. 1 were misinterpreted as they were not connected. At the step of hypothesis verification, 69.7% of incorrect hypotheses were rejected, while all correct hypotheses were preserved. At the step of selection, nine correct hypotheses were erroneously rejected because: (1) an incorrect physical object was closer to a string which represented a label of other physical object, (2) although a single string represented labels of two physical objects, only one physical object was selected. We consider that these errors indicate the limitations of our method which interprets the physical structure. In order to recover these errors, it is necessary to introduce the analysis of linguistic meaning of strings instead of the selection step.

## 7 Conclusion

We have presented a method of understanding conceptual diagrams. To cope with the local ambiguities in interpretation of physical structure, we utilize the technique of hypothesis generation and verification. From the experimental results for 50 samples of conceptual diagrams, we have confirmed that our method is effective but has some limitations of interpretation. The remaining work is the interpretation of conceptual diagrams from their images, and incorporation of natural language processing to improve the accuracy.

## References

[1] K. Tombre. Technical Drawing Recognition and Understanding: From Pixels to Semantics. *Proc. of IAPR Workshop on MVA'92*, pp.393–402, 1992.

[2] Y. Nakamura, R. Furukawa and M. Nagao. Diagram Understanding Utilizing Natural Language Text. *Proc. of the 2nd Int'l Conf. on Document Analysis and Recognition*, pp.614–618, 1993.

[3] R. Kasturi, S. T. Bow, W. El-Masri, J. Shah, J. R. Gattiker and U. B. Mokate. A System for Interpretation of Line Drawings. *IEEE Trans. PAMI*, Vol.12, No.10, pp.978–992, 1990.