J. Kansai Soc. N. A., Japan, No. 212, September 1989

Perception of Picture Interference and its Application to Expert System for Shipboard Equipment Layout (ESSEL)*

By Kazuhiko HASEGAWA (*Member*),** Mato HATTORI (*Student member*),*** Takuji UEMURA (*Member*),**** and Tomoyoshi TANABE*****

A methodology how to deal with pictures or to perceive their interference in computer systems is discussed. The proposed method can recognize not only various situations between two pictures, but also more spatial information between them. These functions to manipulate pictures are useful for some applications such as an intelligent CAD system and any computer systems in which pictures must be handled or perceived. The method is applied to an automatic layout design tool named Expert System for Shipboard Equipment Layout (**ESSEL**).

1. Introduction

Picture perception or handling is belonging to one of the "deep knowledges" in the field of knowledge engineering. We can easily perceive and handle pictures in our daily life, but we cannot say the procedure to do so clearly. There must be some kind of common sense or visual understanding. In this paper we are proposing a method to perceive pictures and their interference by well-known geometrical relations. We also propose a new concept of *spatial distance* between two pictures which must be a key in computerization of layout design.

We will show an example to explain what is the very point to be discussed and treated in a layout problem, when we use a computer for this purpose. Suppose that you are doing jobs on your desk which is almost occupied by piles of manuscripts, reports, letters etc. Now your secretary serves you a cup of coffee. There happens two cases then. You move some piles of papers aside, or your secretary tries to find any space to put the cup on. In either case, both of you know where to move or where to put at a glance without any measurements. How do you understand the space?

There is an approach called image processing. This technique is already developed and utilized for many engineering and industrial applications such as remote sensing photo processing and automatic faults detection in a production line. As main processes used in this method are bit-by-bit calculations such as addition, subtraction, differential, correlation etc., it can detect, for instance, a small flaw on the print pattern of a circuit board, but it cannot say what the flaw is or where it is by itself. Maybe the operator should judge by watching the monitor TV. We need to develop the mehtod to answer these questions directly.

For these purposes, fuzzy theory or neuro-computing may be applicable. However, we cannot specify what data or characteristics should be extracted from the original data of pictures.

2. Pictures in layout problems

2.1 Relations between pictures and objects

First of all, we may specify the problem to be dealt with. All pictures are assumed to exist in a plane, or a plane figure. So each picture represents a 2D object directly or a plane figure of a 3D object. Each object is, for example, such as a desk or a chair. We deal with picture interference in a plane, or we assume the dimension of each object is 2.5 (a plane figure plus height). These relations are shown in Fig. 1. So we can treat 2D and 2.5D layout problems. An example of 2D layout is so-called "nesting", which means the efficient layout problem for cutting parts from a steel plate etc. An example of 2.5D problem is the layout of furniture. We don't treat a complete 3D layout problem such as packing. However, most of 3D layout problem may be treated as 2D layout problem by slicing the space at a certain interval of height.

---

* Read at the Spring Meeting of the Kansai Society of Naval Architects, Japan, May 25, 1989

** Osaka University
*** Graduate School, Osaka University
**** Hitachi Zosen Marine-Tech Co.
***** Kawasaki Heavy Industries, Ltd.

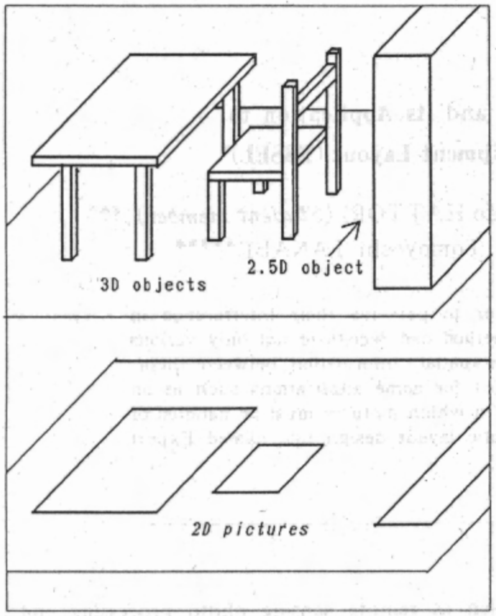

Fig. 1 Relations between pictures and objects

2.2 Picture representation

There are many ways to represents a 2D picture. Definition of a picture and its data structure should be carefully selected, because they may affect the program style and efficiency very much. The most common and simplest way is to represent by a polygon. We should further restrict that the polygon does not have any holes in it and each side of the polygon does not cross or touch each other. Next problem is how to treat a polygon in a computer system. We can define a polygon by a series of 2D points, 2D line segments or a closed area of a series of 2D lines. As a line segment is defined by a pair of 2D points, definitions by 2D points and 2D line segments are equivalent. So, we define a polygon by a series of 2D points here. But we should add a few data to denote which direction is the *front* of the polygon, where is the *center of rotation* and what is the *representing length*. These relations are shown in Fig. 2.

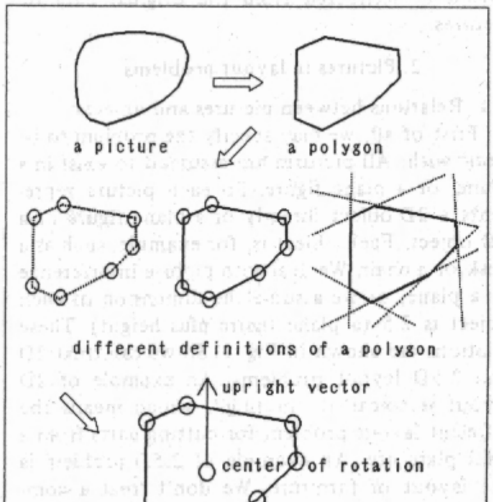

Fig. 2 Definition of a polygon

2.3 Functions for picture interference

The most basic functions to manipulate pictures are to calculate the length between two points, to calculate the angle between two vectors and to obtain the point where two lines cross. These functions are all very easy, so we neet not explain any more except the following points. First, we should take care of the definition of angle. We treat angle between $\pm 180°$. Next, before obtaining the crossing point of two lines, we should check whether two lines are parallel. These three basic functions are essential for almost all functions to check picture (polygon) interference.

We will define the following symbols (cf. Fig. 3).

$|\mathrm{AB}|$: length between points A and B

$\overline{\mathrm{AB}}$: straight line passing points A and B

$\overrightarrow{\mathrm{AB}}$: vector directed from A to B

$\overset{\frown}{\mathrm{AB}}$: line segment whose ends are A and B

$\angle \mathrm{APB}$: angle from $\overrightarrow{\mathrm{PA}}$ to $\overrightarrow{\mathrm{PB}}$

$\{V_k\}$: polygon specified by vertices $V_1, V_2, \ldots$ and V_k

With the combination of the basic functions we can check whether an arbitrary point P is on a certain line $\overline{\mathrm{AB}}$ or not. As any line can be specified by two points A and B, we can rewrite the problem as to compare the angle between two vectors $\overrightarrow{\mathrm{PA}}$ and $\overrightarrow{\mathrm{PB}}$ with zero. If the angle is zero, the point P is on the line $\overline{\mathrm{AB}}$, but not between A and B. If no, we then compare the length between two points A and B with the sum of the lengths $|\mathrm{PA}|$ and $|\mathrm{PB}|$. If both are equal, the point P is on the line segment $\overset{\frown}{\mathrm{AB}}$. If no again, the point P is not on the line $\overline{\mathrm{AB}}$. We can write these relations as follows.

$$\angle \mathrm{APB} = 0 \quad \rightarrow \text{ P is on } \overline{\mathrm{AB}} \quad \cdots\cdots(1)$$

$$|\mathrm{AB}| = |\mathrm{AP}| + |\mathrm{BP}| \rightarrow \text{ P is on } \overset{\frown}{\mathrm{AB}} \quad \cdots\cdots(2)$$

In a layout problem, sometimes we want to set an object so as just to touch to a certain boundary or to be adjacent to another object. Using the proposition (2), we can check that two line segments touch each other at the end points

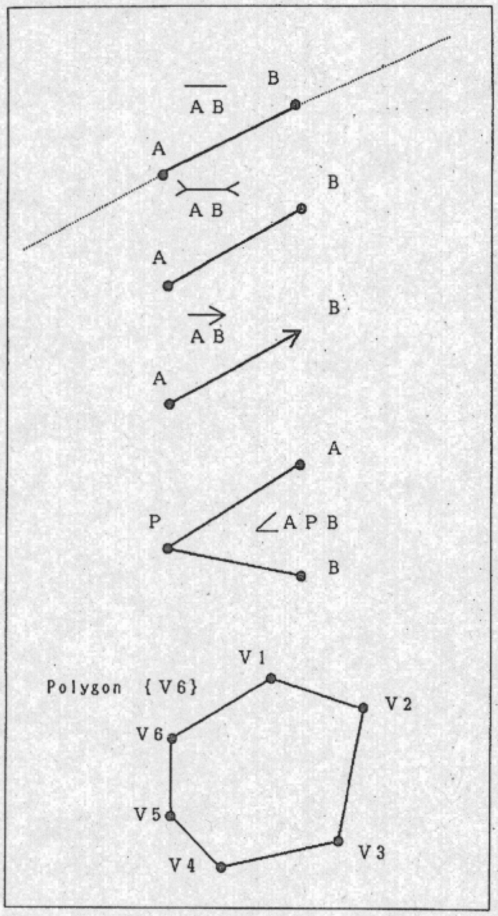

Fig.3 Symbols definition

of the segments. The proposition can be written as follows.

C is on $\overleftrightarrow{AB}$,
D is on $\overleftrightarrow{AB}$,
A is on $\overleftrightarrow{CD}$, ·····(3)
or B is on $\overleftrightarrow{CD}$
→ $\overleftrightarrow{AB}$ and $\overleftrightarrow{CD}$ are touching.

As same as above, we can write a proposition to check two line segments crossing each other in the range of the segments as follows.

$\overline{AB}$ crosses with $\overline{CD}$ at P,
AB and $\overleftrightarrow{CD}$ are not touching,
P is on $\overleftrightarrow{AB}$, ·····(4)
and P is on $\overleftrightarrow{CD}$
→ $\overleftrightarrow{AB}$ and $\overleftrightarrow{CD}$ are crossing.

The second term of proposition (4), which can be read that two segments must not be touching, is intended to identify the crossing from the touching.

Next, we will provide following propositions to check interference between a point and a polygon.

for every i in k
Sum of $\angle V_iPV_{i+1} = 360°$
→ P is including in $\{V_k\}$ ·····(5)

for any i in k
P is on $\overleftrightarrow{V_iV_{i+1}}$ → P is on $\{V_k\}$ ·····(6)

Now, we can get the proposition to check whether two polygons are crossing each other.

for any i in k and any j in k'
$\overleftrightarrow{V_iV_{i+1}}$ and $\overleftrightarrow{U_jU_{j+1}}$ are crossing ·····(7)
→ $\{V_k\}$ and $\{U_{k'}\}$ are crossing.

Left side of proposition (7) is not necessary condition but sufficient condition to say two polygons are crossing (cf. Fig. 4a). In some special cases such as Fig. 4b, we must further check whether each plygon includes at least one vertex of the other polygon, while there is at least one vertex *not included* . Exact proposition to check crossing between two polygons is as follows.

for any i, i' in k and any j, j' in k'
$\overleftrightarrow{V_iV_{i+1}}$ and $\overleftrightarrow{U_jU_{j+1}}$ are crossing
or($\{V_k\}$ includes U_j
and $\{V_k\}$ does not include $U_{j'}$)
or ($\{U_{k'}\}$ includes V_i ·····(8)
and $\{U_{k'}\}$ does not include $V_{i'}$)
→ $\{V_k\}$ and $\{U_{k'}\}$ are crossing.

Using these propositions, we can provide the propositions to check interference between two polygons as follows.

$\{V_k\}$ and $\{U_{k'}\}$ are not crossing
and for any i in k and any j in k' ·····(9)
$\overleftrightarrow{V_iV_{i+1}}$ and $\overleftrightarrow{U_jU_{j+1}}$ are touching
→ $\{V_k\}$ and $\{U_{k'}\}$ are touching.

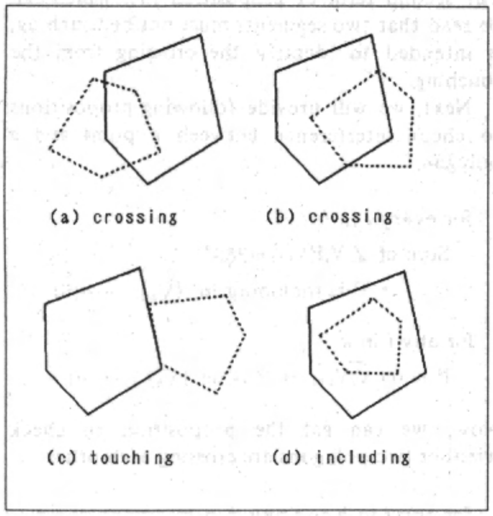

Fig.4 Various situations between two polygons

$\{V_k\}$ and $\{U_{k'}\}$ are not crossing
and for any i in k and every j in k'
V_i is not on $\overleftrightarrow{U_jU_{j+1}}$ .....(10)
and V_i is included in $\{U_{k'}\}$
$\rightarrow$ $\{V_k\}$ is including in $\{U_{k'}\}$.

3. Spatial perception of picture interference

Using the above mentioned functions for picture interference, we have developed the pilot system of automatic layout design tool called **"ESSEL"** as described in the next section. In this system, it can be conscious of picture interference such that the considering two pictures are crossing, touching or not. Your computerized secretary now can find the space for the cup by testing – putting the cup temporarily and checking the interference between other objects, and if it touches or crosses, she tries it again. In our system **"ESSEL"** these behaviours can be seen when the mooring winch for breast lines are set with enough distance from the windlass, or deck side fittings such as closed chocks are set at appropriate positions so as to touch the deck side line.

However, we cannot say how much the considering pictures interfere or what condition is *'not crossing'* or *'not touching'* yet. In this section, we try to answer these questions.

3.1 Distance between two pictures not crossing each other

Suppose the situations as in Fig. 5. There are four cases, and the distance of each center of pictures is all the same. But, we may feel that pictures in Fig. 5a or Fig. 5b are closest, then

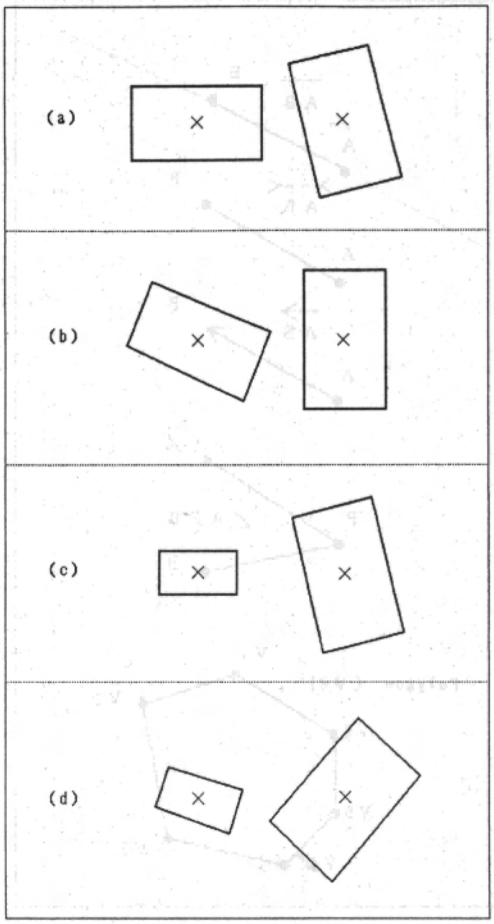

Fig.5 Distance between two pictures
– effect of size and direction

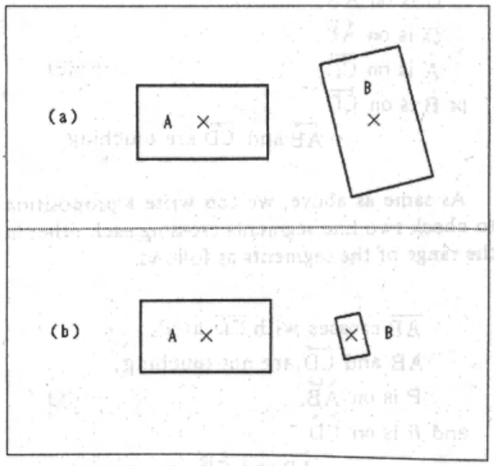

Fig.6 Distance between two pictures
– effect of picture size

those in Fig. 5d and those in Fig. 5c are least close. So, it is clear that the distance of each center cannot be the index. Next, we can try the distance between the closest vertices of each picture, or the distance of the closest points of each picture. The latter index does not benefit, because it takes much time in calculation. The former factor is easy to calculate, but cannot distinguish the difference between Fig. 5a and Fig. 5d., and it may misjudge the pictures in Fig. 5a are closer each other than those in Fig. 5b. We will show another example as in Fig. 6. We also feel pictures in Fig. 6a is closer than those in Fig. 6b. In Fig. 6b, sometimes we feel for picture A, picture B is very close, but for picture B, picture A is not so close. Maybe, we feel distance by different scale according to each picture.

3.2 Expansion of the concept of line segment crossing to measure the distance

In 2.3, we have introduced a method to check if two line segments cross each other or not. In this method we used the angle to check it, but there is another method using vectors. Consider two lines defined by two points A, B and C, D respectively are crossing at a point P (see Fig. 7a). We already omit the case that two lines are parallel. Then we can write the vector $\overrightarrow{OP}$ (O is the origin) as

$$\overrightarrow{OP}=\overrightarrow{OA}+\mu_{AC}\overrightarrow{AB} \quad \text{or} \quad \overrightarrow{OP}=\overrightarrow{OC}+\mu_{CA}\overrightarrow{CD} \qquad \cdots\cdots(11)$$

where μ_{AC} is the mediation parameter of vector $\overrightarrow{AB}$ by vector $\overrightarrow{CD}$ and μ_{CA} is *vice versa* and if

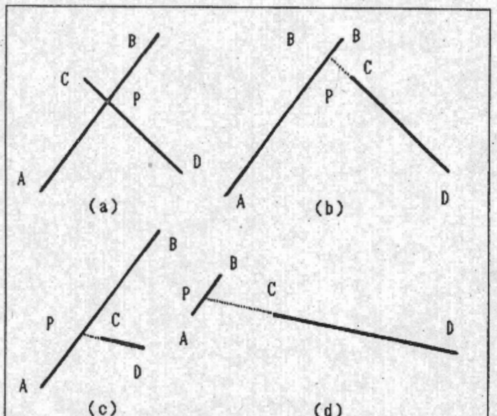

Fig. 7 Spatial distance between two line segments

$$\text{and} \quad 0\leq\mu_{AC}\leq 1, \quad 0\leq\mu_{CA}\leq 1 \qquad \cdots\cdots(12)$$

are satisfied, the point P is on the line segments $\overleftrightarrow{AB}$ and $\overleftrightarrow{CD}$, so two line segments $\overleftrightarrow{AB}$ and $\overleftrightarrow{CD}$ are crossing each other. Usually we check only the condition (12), but we are now interested in other cases than (12). If either μ_{AC} or μ_{CA} nearly satisfies the condition (12), the two line segments $\overleftrightarrow{AB}$ and $\overleftrightarrow{CD}$ are near each other (Fig. 7b). Besides, the factors μ_{AC} and μ_{CA} are not absolute values, but relative values to compare with the length of each line segment. So if, for instance, $|AB| = 10$, $\mu_{AC} = 0.4$, $|CD| = 2$ and $\mu_{CA} = -0.3$, we can imagine that the line segment $\overleftrightarrow{CD}$ is just beside the line segment $\overleftrightarrow{AB}$ (see Fig. 7c). Contrary, if $|AB| = 2$, $\mu_{AC} = 0.4$, $|CD| = 10$ and $\mu_{CA} = -0.3$ (Fig. 7d), the real distance is far from Fig. 7c, but the psychological distance from the line segment $\overleftrightarrow{CD}$ is about equal to the case in Fig. 7c.

3.3 Spatial distance between two pictures

We can expand the concept of *spacious distance* between two line segments to the case in two pictures. Let us consider a case as in Fig. 8. Each line segment of each picture has the same number of crossing points as the vertices of the opposite polygon, if we include crossing points such as P_1, P_2 and P_4 in the case of the line segment $\overleftrightarrow{AB}$ as shown in Fig. 8. In this case, we can judge as follows:

$\overleftrightarrow{AB}$ is near and facing to $\overleftrightarrow{GH}$ at P_3,
near to $\overleftrightarrow{EF}$ and $\overleftrightarrow{HE}$,
and far from $\overleftrightarrow{FG}$.

We then remember the nearest point P_3 for the line segment $\overleftrightarrow{AB}$ and $\overleftrightarrow{GH}$. Similarly, we can pick up the nearest points for each line segment respectively as Q_3 for $\overleftrightarrow{BC}$, R_1 for $\overleftrightarrow{CD}$, S_4 for $\overleftrightarrow{DA}$ and $\overleftrightarrow{HE}$, P_1 for $\overleftrightarrow{EF}$, and S_2 for $\overleftrightarrow{FG}$. Next, we choose a set of nearest line segments as $\overleftrightarrow{DA}$ and $\overleftrightarrow{HE}$ from the picture ABCD, and $\overleftrightarrow{GH}$ and $\overleftrightarrow{AB}$ from the picture EFGH respectively in this case. Now we can judge that the picture ABCD is about a third distance of its *representing length* apart from the picture EFGH, and the picture EFGH is about a half distance of its *representing length* apart from the picture ABCD. These procedures can be written as follows:

for every i in k and j in k'

if (NOT $0<\mu_{ij}<1$) and $0<\mu_{ji}<1$

$$\mu_{imin}^{jmin}=\min_i\{\min_j\{|\mu_{ij}-0.5|\}\} \qquad \cdots\cdots(13)$$

if $0<\mu_{ij}<1$ and (NOT $0<\mu_{ji}<1$)

$$\mu_{jmin'}^{imin'}=\min_j\{\min_i\{|\mu_{ji}-0.5|\}\}$$

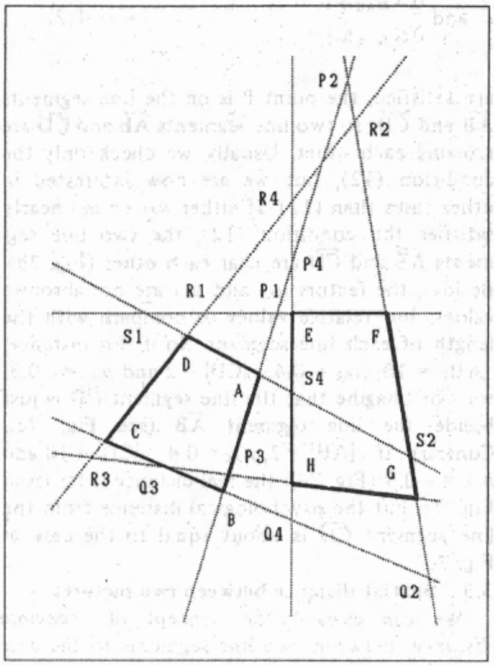

Fig.8 Spatial distance between two polygons

where $imin, imin' \in \{i\}$ and $jmin, jmin' \in \{j\}$ we define the spatial distance from $\{V_k\}$ to $\{U_{k'}\}$ as $\tilde{d}^{\{V_{k'}\}}_{\{U_{k'}\}}$ and the representing length of $\{V_k\}$ as l_k,

$$\tilde{d}^{\{V_{k'}\}}_{\{U_{k'}\}} = \mu^{jmin}_{imin} \frac{|V_{imin} V_{imin+1}|}{l_k}$$

and in the similar way ·······(14)

$$\tilde{d}^{\{U_{k'}\}}_{\{V_k\}} = \mu^{imin'}_{jmin'} \frac{|U_{jmin'} U_{jmin'+1}|}{l_{k'}}$$

where $V_{imin+1} = V_1$ and $U_{jmin'+1} = U_1$ for $imin = k$ and $jmin' = k'$

4. Expert System for Shipboard Equipment Layout "ESSEL"

Recently, AI (artificial intelligence) technology has rapidly developed and been utilized in diagnosis, consultation, monitoring and control problems. The most classic and popular problems which AI researchers had first challenged are puzzles such as "chess", "farmer's dilemma" etc. The key technologies developed through these researches are expression of knowledge such as *frame* and *object*, pattern matching and conflict resolution, inference engine, back tracking and so on. A computer system which solves a problem by rules extracted from experts such as medical doctors and high-level chess players is called *expert system*. We can now get many software products or tools to customize our own expert systems. However, we still lack methods to describe or treat most of human-oriented abilities such as learning, association, inspiration, creation and oblivion.

Designing is purely a human activity and experts sometimes cannot explain or aren't conscious of their behaviours even by themselves. Perception of picture interference is of course one of these veilable concepts. In this section, an approach of expert system for layout design or arrangement is described.

4.1 Shipboard mooring equipments

The basic configuration and terminology of mooring arrangement and mooring equipments on board are shown in Figs. 9 and 10[1]. The problem here is to arrange the required mooring equipments such as windlasses, drums, chocks etc. to the proper position respectively. There are few restriction such as ship classification societies' rules or owners requirements, but designers should pay attentions to easiness of handling, operation and traffic, and check that the rope paths do not cross other deck fittings. Of course, each equipment should not interfere with the others.

4.2 Configuration of "ESSEL"

"ESSEL" is an automatic layout design tool for shipboard equipments for mooring, using an expert system tool OPS83. In this research we apply the system to the layout design in the forecastle deck of an oil tanker. The system has rule base which is made of 7 groups of rules. A group of process control rules **'ks0'** supervises the system flow from **'ks1'** to **'ks4'**, which are shown in Fig. 11, including the first and the last stage. In the first stage, the outline of the f'cle deck, the shape and location of the hawse pipes, the fore mast and, Suez search hatch, and the type of windlasses and winches are given prior. In the following stages, if some fittings are required by some rules, they are automatically picked up from the catalog base. Every movement of the system is visible on a color graphic terminal and every equipment changes its color according to the process state, so the operator can easily monitor the system in real time.

A group of rules **'ks5'** will check the picture interference using the method described in the former section. If there are any interferences, it will try another layout, or send a warning when

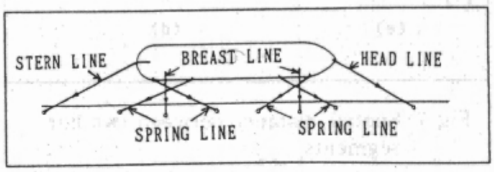

Fig.9 Mooring arrangement

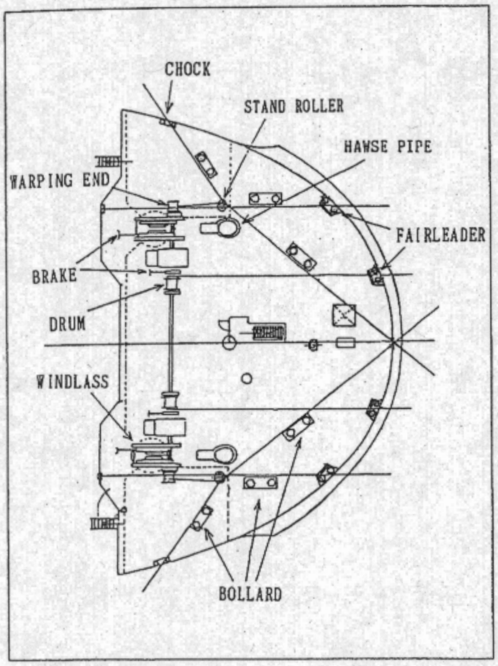

Fig.10 Mooring equipments on a f'cle deck

it doesn't know how to do just in Fig. 11f (where the working area of the winch is crossing the deck end).

4.3 Family of objects

There is a remaining group of rules **'ks6'**, which is not supervised by **'ks0'**, but gets the system initiative temporarily from **'ks0'** when either of groups from **'ks1'** to **'ks5'** requests. This is a group of rules which handles a concept of family of objects.

A concept of family of objects is necessary because of one of the unvisible rules used by human. For example, let us consider a case as in Fig. 12a, where one rope path crosses the hawse pipe. The system checks the interference and suggests to move the stand roller forth (Fig. 12b). But the rope lines and bollards are remaining! Human never says, but the *related objects* should be, of course, moved according to the movement of the stand roller. Now there is a problem. What do *related objects* mean? Apparently they don't mean all related objects, because the warping end etc. need not move. We have introduced the concept of family of objects then. In this case, the family is expressed as in Fig. 13, and the rule is written as

> If an object in a family is moved,
> then all objects *younger* than it
> should be again set according to
> the appropriate rules.

We also prepare the similar rules to move its *children*, its *father* and all the *family* and gather

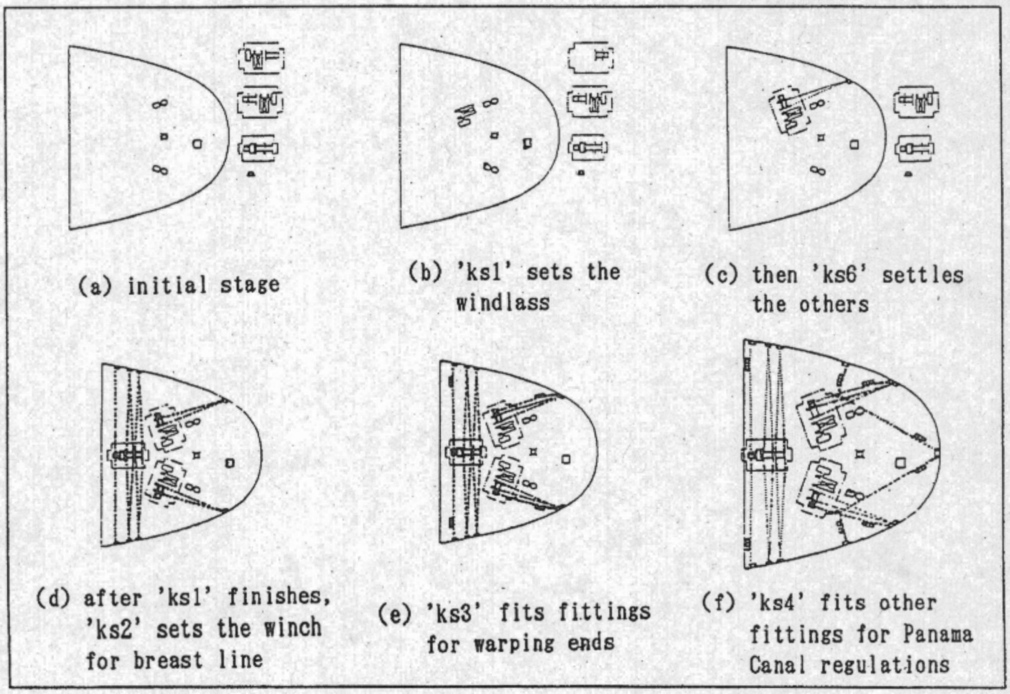

Fig.11 System flow from **'ks1'** to **ks4'**

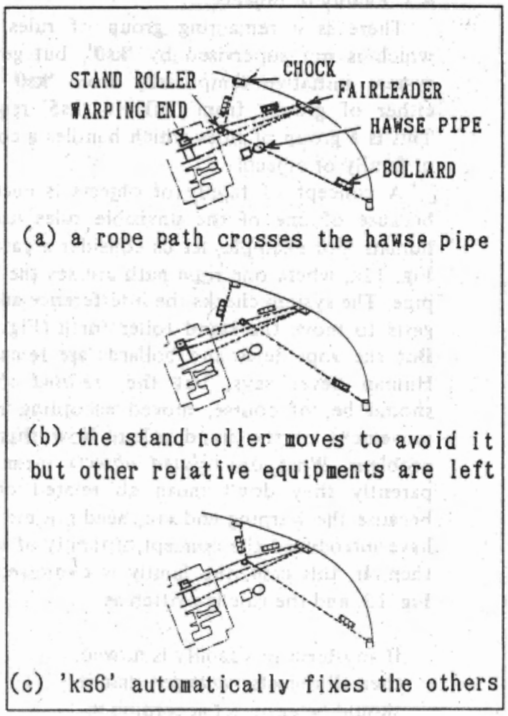

Fig. 12 Movements in **'ks 6'** family of objects

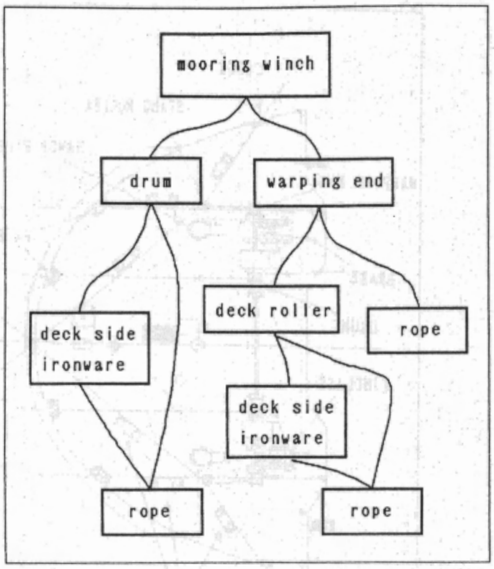

Fig. 13 Hierarchy relation of objects

them into **'ks6'**. With utilizing this group of rules, we need no more specify unnecessarily complex actions in an individual rule, but only describe just as designers do. Then the system successfully changes *related objects* as in Fig. 12c.

4.4 Some results of the system

The benefit of the expert system is flexibility and expansibility. As far as we prepare suitable rules, the system will select the proper rules for the given problem and act as if designers were doing. We here show some results in Fig. 14, where the deck size and the total required force for mooring are the same as in the case of Fig. 11, but the location of hawse pipes and the configuration of windlasses and/or drums are different. The system works satisfactorily in each case and we can see the effect of the variation of winches etc. very easily. Hence, though this system is a proto-type and should be revised more and more, we have the confidence that this type of automatic layout system is quite benefitable in the initial stage of ship design.

5. Conclusions

(1) The proposed method to perceive the picture interfere is very simple and useful for the computer system.

(2) New concepts of *spatial distance* and *family of objects* are proposed.

(3) The expert system of layout design works satisfactorily, utilizing the proposed methods to perceive picture interference.

We will further investigate the method to perceive "space", "beauty of layout" or "designers' sense" in a layout problem. **"ESSEL"** is now under upgrading into **"VESSEL"** as a reult.

This research was conducted as one of research topics in FY 1987 of W.G.2, First Comm., SR210 project of Japan Ship Research

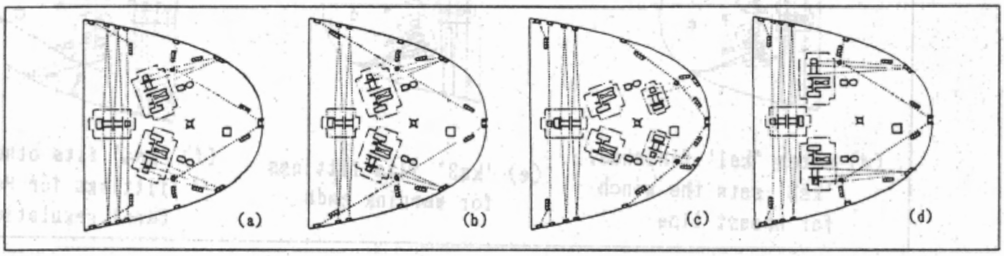

Fig. 14 Some different results of **"ESSEL"** in various initial conditions

Association. The authors were always encouraged by important discussions and valuable suggestions at the meetings. They would like to thank to all the members for their contributions.

Reference

1) W.G.2, Ship Design Comm., S.N.A.J.: "Design Guide for the Mooring of Large Ships (JSDS-4)", Kaibundou, 1971, Kobe.

[Discussion] Dr. Norihiko Matsumoto (NKK)
I think some of objects are not necessarily governed by your concept "family of objects". Isn't it always necessary?

[Author's reply] In this paper all objects are connected by the concept "family of objects", even if any of them is not related with any other objects. Instead, as is described in 4.3, we prepare various rules to manipulate its children and descendants, its father's children (its brothers) and descendants, its father and ancestor and so on. If the object is independent, we only do not call these rules on the right-hand side of the very rule which manipulates this object.

[Discussion] Prof. Keiichi Karasuno (Hokkaidou University)
Is the technique of pattern recognition available to understand pictures?

[Answer] Partially, yes. But, the main procedure used in pattern recognition is image processing. As is described in 1., image processing requires huge memory. Besides, this technique is necessary, only when input image including pictures is offered as "image". In our paper, input image itself does not exist. We need not visible image at all. We use it only for output to help human operator. The proposed method is still necessary after the technique of pattern recognition extracts some characteristics such as edge lines of pictures from raw image.