Symbolic Representation and Reasoning for Rectangles with Superposition

Takako Konishi
Graduate School of Science & Technology
Kwansei Gakuin University
2-1, Gakuen, Sanda, 669-1337, JAPAN
Email: t.konishi@kwansei.ac.jp

Kazuko Takahashi
School of Science & Technology
Kwansei Gakuin University
2-1, Gakuen, Sanda, 669-1337, JAPAN
Email: ktaka@kwansei.ac.jp

Abstract—**This paper discusses the superposition of qualitative rectangles so that some parts are visible and other parts are hidden based on the user's requirements. Qualitative rectangles are rectangles whose size and edge ratios are not fixed. We investigate the conditions under which such a superposition succeeds as well as the manner in which such superposition occurs. We also show an algorithm for superposing a multiple number of qualitative rectangles. It is applicable to construct qualitative spatial database for multiple window placement systems.**

***Keywords*-qualitative knowledge representation; rectangle packing; spatial database.**

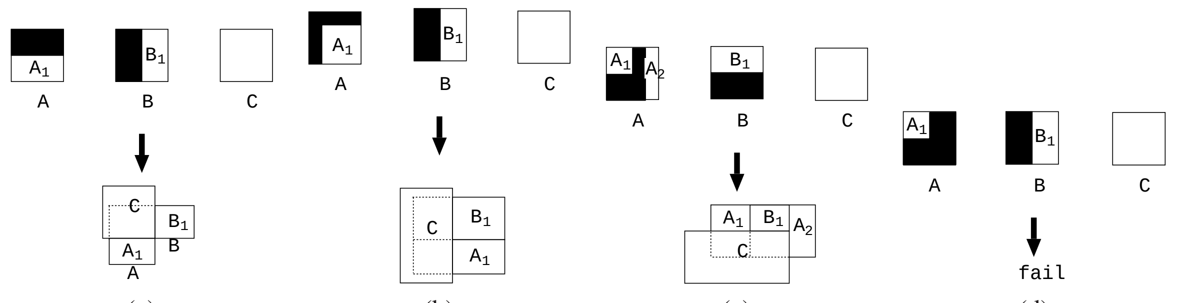
I. Introduction

This work was inspired by an issue which occurs during multiple window placement. When working on PCs, we often open multiple windows in a superposed manner within the restricted space of a monitor. At that time, we seldom need all the content displayed in the windows; rather, we pick up the necessary portions by frequent use of mouse operations, such as clicking or dragging. Efficient placement of windows such that only the necessary parts are visible could serve as a useful tool to reduce our annoying mouse operations. To achieve this, we should specify the parts of each window to be visible and find the superposition of the windows that satisfies the specification as well as the simple positional relationship on the two-dimensional plane. In general, efficient placement of objects has been studied as a type of packing problem to which an optimal solution is searched [1]. Much work has been undertaken in several application areas, such as VLSI design [2], label-placement problems [3], [4] and more. In these studies, the problem of how multiple objects are located in a two-dimensional plane has been approached under circumstances not involving superposition. To the best of the authors' knowledge, no study has been performed on the location of objects with superposition.

In this study, we discuss rectangle placement with superposition. We treat rectangles using qualitative representation: their sizes and the ratios of their edges are unfixed. In each rectangle, the desired visible part of a rectangle is specified. We discuss a manner of superposing them so that all desired visible parts are in the foreground and all desired hidden parts are in the background. Figure 1 illustrates several examples. Assume that three rectangles A, B, and C are given with a requirement of visibility specified by a user. The white indicates the parts that one wants to be visible, and the black indicates the parts that one wants to be hidden. In this figure, (a), (b), and (c) are successful cases, whereas (d) is not. In (c), first reduce B's width to fit the vertically-long-size subpart of the black part of A, then C is put on the black part in the lower left part of the resultant figure. In (d), visible black parts remain after superposing A and B cannot be hidden by C in any superposition of A and B. In this paper, we show how we evaluate the success of superposition and placement in these cases.

Considering multiple window placement, a window is divided into several frames in most application software and their dividing patterns are restricted. An automatic window placement can be accomplished by the following mechanism: add the attribute value on visibility to each frame of each window, store in the database the list of pairs of a combination of multiple windows with the attribute values and their best placement; retrieve the best placement from the database for the combination of windows on their invocation, and display them. Attribute value on each frame can be decided through learning from lots of examples, however, this issue is beyond the scope of this paper. Here, we assume that it is given in advance and discuss reasoning about the best placement for a given combination of windows.

We take a qualitative approach. One reason for this is that it enables symbolic handling of objects. In general, spatial data can be inconveniently large to store and handle. Symbolic handling reduces this computational complexity. Another reason is that it is enough to know the relative positional relationship of objects on a two-dimensional plane and their foreground/background relationship, ignoring the exact size or position of each object. Such an idea is considered to be a type of qualitative spatial reasoning (QSR) in the field of artificial intelligence [5], [6], [7], [8].



(a) (b) (c) (d)

Figure 1. Examples of superposing rectangles

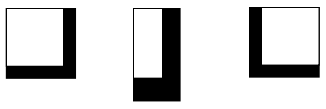
There are several studies on qualitative spatial database. For example, Wang and Liu showed an application of QSR to geospatial semantic web by constructing qualitative spatial database, in which objects and their qualitative relations are stored instead of coordinates, from the Geography Markup Language (GML) [9]. Santos and Amaral proposed an approach to make qualitative database, by introducing qualitative identifier such as direction and relative distance, and apply it to data mining [10]. Although these studies showed effectiveness of qualitaive spatial database, further studies are required. This paper aims at enlarging possible application areas of qualitative spatial database.

This paper is organized as follows. In Section II, we define the target object and describe its qualitative representation. In Section III, we describe the operations of superposing a pair of qualitative rectangles, and discuss reasoning about superposition. In Section IV, we discuss the result of superposition and show an algorithm for superposing multiple qualitative rectangles. Finally, in Section V, we show the conclusion.

II. Description

We call a superposing entity *a unit*. A unit is divided into `WHITE` which should be visible, and `BLACK`, which should be hidden. `BLACK` is divided into a *core region* and a *non-core region*, which will be defined later. The outer side of a unit is called `GRAY`. The length of edges and the ratios of a unit and of each region are unfixed. On the other hand, the orientation of a unit should be fixed. We only use rectangles situated in an upright position and do not consider those in an inclined orientation. These means that (a) and (b) in Figure 2 are regarded as equivalent, while (a) and (c) are regarded as different.

Each connected `WHITE` is called *a white region*, the core region and each connected non-core region are called *black regions*, and `GRAY` is called *a gray region*. The white, black, and gray regions have attribute values related to visibility, denoted by 'w,' 'b,' and 'g.' 'w' and 'b' denote that the region should be visible and hidden, respectively. 'g' denotes there is no requirement with respect to visibility.



(a) (b) (c)

Figure 2. Qualitative rectangles

Considering a structure of frames of WEB pages or a style of dividing a window into sub-windows in many applications, we restrict the type of unit to those in Figure 4. Any unit can be defined as a qualitative rectangle obtained by the following operation that fits black plates into a white rectangle. Conversely, a qualitative rectangle obtained by this operation is only the units shown in Figure 4. Let $a * b$ represent a size of a unit whose length is a and height is b. Consider the two black plates whose sizes are $x * b$ ($0 \leq x \leq a$) and $a * y$ ($0 \leq y \leq b$). Fit these plates into a white rectangle while preserving their orientations in either of the following manners. Symbols enclosed in parentheses denote names of unit types.

(0) No plate is fit (W).

(1) Only one of the plates is fit (B, I1, I2).

(2) Both plates are fit (L1, T1, PLUS).

(3) Extend L1 and T1, respectively, where the white region is added to the part on which the edge of a size a or b is connected to the outer part (L2, T2).

Definition 1. *The unit obtained in this manner is said to be* valid.

Types I1 and I2 are called *straight-plate-unit*s. Types L1, L2, T1, T2, and PLUS are called *cross-plates-unit*s. All units of the same type are called *a pattern*.

For all units other than the W-type unit, the *core region* is defined. For B-type and straight-plate-units, the core region is defined as the entire `BLACK`. For cross-plates-units, the core region is defined as the intersection of the two plates, and the region not included in the core region is called the *non-core region* (Figure 3).

In a valid unit, all white regions are convex, and there

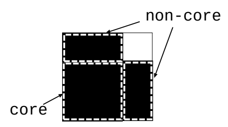


Figure 3. Core region and non-core region of cross-plates-unit

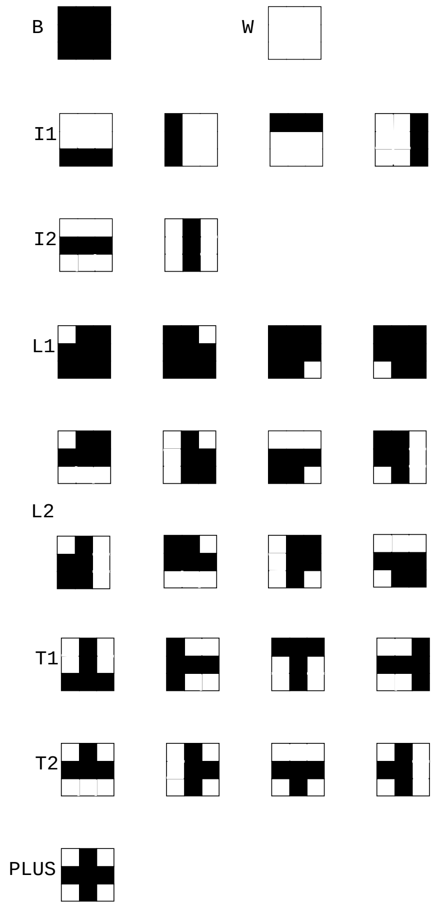


Figure 4. All units

exists only one connected BLACK.

We denote the core region of a unit X by $Core_X$. The valid unit can be uniquely represented as a quadruple of attribute values composed of $Core_X$'s upper region, right region, lower region, and left region. We call this representation *a representation for a unit*. For example, the representation for a unit in Figure 3 is $\langle b, b, g, g \rangle$, since the core region has black regions in its upper side and right side, whereas it is connected to the outside in its lower side and left side. Note that the positional relationships of regions are preserved even if the size of a unit is changed.

Let V, R and $TYPE$ indicate a set of attribute values, a set of representations for units, and a set of types, that is:

$V = \{b, w, g\}$

$R = \{\langle r_1, r_2, r_3, r_4 \rangle \mid r_i \in V (i = 1, \ldots, 4)\}$

$TYPE = \{$'B','W','I1','I2','L1','L2','T1','T2','PLUS'$\}$

Then, the function $type$ that defines the type for a representation $r \in R$ is defined as follows.

$type : R \rightarrow TYPE$

$type(\langle g, g, g, g \rangle)$	= 'B'
$type(\langle w, w, w, w \rangle)$	= 'W'
$type(\langle w, g, g, g \rangle)$	= 'I1'
$type(\langle w, g, w, g \rangle)$	= 'I2'
$type(\langle b, g, g, b \rangle)$	= 'L1'
$type(\langle b, g, w, b \rangle)$	= 'L2'
$type(\langle b, b, g, b \rangle)$	= 'T1'
$type(\langle b, b, w, b \rangle)$	= 'T2'
$type(\langle b, b, b, b \rangle)$	= 'PLUS'

Note that type W is defined with the assumption that there exists a tiny core region surrounded by white regions, as $Core_X$ does not exist.

The projections from $r \in R$ to its elements are defined as follows.

$up/dn/lt/rt : R \rightarrow V$

Let r be $\langle r_1, r_2, r_3, r_4 \rangle$.

$up(r)$	$= r_1$
$rt(r)$	$= r_2$
$dn(r)$	$= r_3$
$lt(r)$	$= r_4$

Moreover, the function $rotate(r)$ that denotes a $\pi/2$ clockwise rotation of a unit r and the function $symm(r)$ that denotes a symmetric transformation of a unit r are defined as follows:

Let r be $\langle r_1, r_2, r_3, r_4 \rangle$.

$rotate : R \rightarrow R$

$rotate(\langle r_1, r_2, r_3, r_4 \rangle) = \langle r_2, r_3, r_4, r_1 \rangle$

$symm : R \rightarrow R$

$symm(\langle r_1, r_2, r_3, r_4 \rangle) = \langle r_1, r_4, r_3, r_2 \rangle$

Note that $type(r) = type(rotate(r))$ and $type(r) = type(symm(r))$ hold.

III. Reasoning about Superposition

A. The principle

When n ($n \geq 3$) units are given, we consider the manner of their superposition in which all white regions are visible and all black regions are hidden.

Here, we place units sequentially. k-th unit ($n \geq k \geq 2$) should be placed on the figure composed of $k-1$ units so that at least one region of the former is placed on at least one region of the latter. That is, we do not consider the placement in which, after two units are placed in a disconnected manner, the third unit is placed onto the black region of the two rectangles simultaneously. Thus, there should be at least one W-type unit, and only one connected rectangular BLACK should be visible when superposition of $n-1$ units is completed.

Here, we assume that there is one W-type unit. When more than one W-type unit exists, the scenario can be considered similarly.

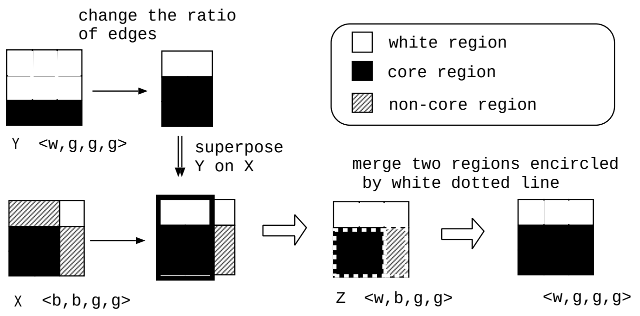


Figure 5. The case in which $merge$ is necessary

B. Superposing the core regions

Definition 2. *Suppose that a unit X and an straight-plate-unit Y are given. Let $Core_X$ and $Core_Y$ be the core regions of X and Y, respectively. The superposition in which $Core_Y$ is placed exactly on $Core_X$ is called* puton operation.

Let Z be the resultant figure of $puton$, and let $Core_Z$ be the superposed region of $Core_X$ and $Core_Y$. We extend a representation for a unit to be available as a representation for Z. *A representation for* Z is a quadruple of the attribute values of visible regions surrounding $Core_Z$.

First, we compute the attribute values of the regions around $Core_Z$. We define the function on, which computes the attribute value of the visible region when one region is placed exactly on another region, from those of the two regions.

$on : V \times V \rightarrow V \cup \{\mathtt{U}\}$
$on(b, b) = \quad b$
$on(b, w) = \quad w$
$on(w, w) = \quad \mathtt{U}$
$on(w, b) = \quad \mathtt{U}$
$on(g, v) = \quad v$ where $v \in V$
$on(v, g) = \quad v$ where $v \in V$

'U' means that the operation is undefined for that case.

It sometimes occurs that X's black regions are visible in Z. If they are connected with $Core_Z$ by lines, it is necessary to merge them to define the merged region as new $Core_Z$. For example, in Figure 5, X and Y are represented as $\langle b, b, g, g\rangle$ and $\langle w, g, g, g\rangle$, respectively. When we place Y on X such that $Core_Y$ is placed on $Core_X$, the resultant figure Z is represented as $\langle w, b, g, g\rangle$. X's non-core region is visible, which is connected with $Core_Z$ by a line. Then, this region is merged with $Core_Z$. This function $merge$ is defined as follows.

Let $r = \langle r_1, r_2, r_3, r_4\rangle$. If it satisfies (c1), then $merge(r)$ succeeds. When $merge$ succeeds, its value is defined as follows.

(c1) $\bigwedge_{i=1,\ldots,4}(r_i \neq \mathtt{U})$.

$merge : R \rightarrow R$

$merge(r) =$

$$\begin{cases} \langle g, r_2, g, r_4\rangle & \text{if } r_1 = b \wedge r_2 \neq b \wedge r_3 = b \wedge r_4 \neq b \\ \langle r_1, g, r_3, g\rangle & \text{if } r_1 \neq b \wedge r_2 = b \wedge r_3 \neq b \wedge r_4 = b \\ \langle g, r_2, r_3, r_4\rangle & \text{if } r_1 = b \wedge r_2 \neq b \wedge r_3 \neq b \wedge r_4 \neq b \\ \langle r_1, g, r_3, r_4\rangle & \text{if } r_1 \neq b \wedge r_2 = b \wedge r_3 \neq b \wedge r_4 \neq b \\ \langle r_1, r_2, g, r_4\rangle & \text{if } r_1 \neq b \wedge r_2 \neq b \wedge r_3 = b \wedge r_4 \neq b \\ \langle r_1, r_2, r_3, g\rangle & \text{if } r_1 \neq b \wedge r_2 \neq b \wedge r_3 \neq b \wedge r_4 = b \\ \langle r_1, r_2, r_3, r_4\rangle & \text{otherwise} \end{cases}$$

Success of $puton$ operation

For representations $r=\langle r_1, r_2, r_3, r_4\rangle$ and $r'=\langle r'_1, r'_2, r'_3, r'_4\rangle$, if $merge(on(r_1, r'_1), on(r_2, r'_2), on(r_3, r'_3), on(r_4, r'_4))$ succeeds, $puton$ succeeds and its value is defined as follows.

$puton : R \times R \rightarrow R$
$puton(r, r') =$
$\quad merge(on(r_1, r'_1), on(r_2, r'_2), on(r_3, r'_3), on(r_4, r'_4))$

The following property clearly holds due to the definition of $puton$.

Theorem 3. *If the $puton$ operation succeeds, Z's* BLACK *is connected, and all of its white regions are convex.*

Next, consider that we superpose the third unit on Z. There are two necessary conditions for this operation to succeed. First, if Z's BLACK is not rectangular, superposing the W-type on Z will not succeed. Second, if Z is not valid, continuous superposition cannot be considered. We describe how to verify these two conditions.

Shape verification of BLACK

Let r be a representation for Z. If r satisfies both (c2) and (c3), then Z's BLACK is rectangular.

(c2) $(up(r) = b \vee dn(r) = b) \rightarrow$
$\quad (lt(r) \neq b \wedge rt(r) \neq b)$

(c3) $(lt(r) = b \vee rt(r) = b) \rightarrow$
$\quad (up(r) \neq b \wedge dn(r) \neq b)$

Definition 4. *For a figure Z obtained by superposing n ($n \geq 2$) units, if there exists only one connected* BLACK *that is visible and rectangular, then Z is said to be* effective.

Shape verification of the whole figure

Let $r = \langle r_1, r_2, r_3, r_4\rangle$ and $r' = \langle r'_1, r'_2, r'_3, r'_4\rangle$ be representations for units X and Y, respectively. For the entire shape of Z to be rectangular, the white region of Y should not be placed on GRAY of X. Therefore, if r and r' satisfy (c4), then the shape of Z is a rectangle.

(c4) If there does not exist i ($1 \leq i \leq 4$) such that $r_i = g, r'_i = w$.

C. Superposition by embedding

We can consider another superposition operation of $embed$.

Definition 5. *If we place the whole unit on the entirety or on a part of* BLACK *of the other unit, this operation is called an* embed operation.

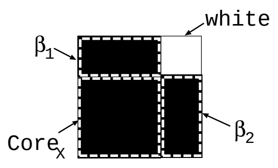


Figure 6. The regions to be hidden in L1

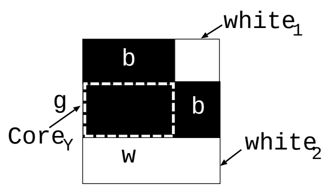


Figure 7. Representation of locations of white regions

IV. Result of superposition

In Definition 2, we defined $puton$ operation for a superposition of a straight-plate-unit and a unit. In this section, we extend this operation for any pair of unit types. And we discuss the effectiveness and validity of the resulting figures of the operation $puton$ and $embed$ for each pair of unit types.

A. Superposition on B/W type

Assume that we superpose some unit on the B-type. The resultant figure is effective if and only if we superpose the straight-plate-unit, and it is valid for any type.

On the other hand, it is impossible to place any unit on the W-type.

B. Superposition of straight-plate-units

Assume that we superpose the straight-plate-unit on the straight-plate-unit. The resultant figure obtained by the $puton$ operation is not always valid, as its entire shape may not be a rectangle. The resultant figure obtained by the $embed$ operation is not always valid, as a white region may not be convex. On both operations, the resultant figure is effective.

C. Superposition of the straight-plate-unit on the cross-plates-unit

Assume that we superpose the straight-plate-unit on the cross-plates-unit. The resultant figure obtained by any operation is not always valid and not always effective. However, the following property holds.

Theorem 6. *When we superpose the straight-plate-unit on the cross-plates-unit, the effective figure cannot be obtained by any operation other than the puton operation.*

Proof:

Consider the $puton$ operation that places a straight-plate-unit Y on an L1-type unit X shown in Figure 6. In this case, `BLACK` is divided into three regions: one core region $Core_X$ and two non-core regions β_1 and β_2. Let $Core_Y$ be Y's core region.

One or two of the $Core_X, \beta_1, \beta_2$ should be hidden so that the resultant superposed figure is effective.

(i) Only one region is hidden.

If only $Core_X$ is hidden, β_1 and β_2, which are disconnected, are visible. Therefore, the result is not effective. If only β_1 is hidden, $Core_X, \beta_2$ and $Core_Y$ are visible in the resultant figure. Considering the relative position of $Core_X, \beta_1$ and β_2, it is impossible to make a rectangle by merging $Core_X, \beta_2$ and $Core_Y$ and to hide β_1 at the same time. Therefore, the result is not effective. Similarly, the result is not effective if only β_2 is hidden.

(ii) Two regions are hidden.

Since β_1 and β_2 are disconnected, they are not simultaneously hidden by a single unit. If both $Core_X$ and β_1 are hidden, β_2 and $Core_Y$ are visible. We must place Y's regions onto both $Core_X$ and β_1 to make them hidden. Moreover, we must make a rectangle by merging β_2 and $Core_Y$. The only place where $Core_Y$ should be placed to satisfy both conditions is $Core_X$, and this placement is identical to the $puton$ operation.

Based on the above analysis, the resultant figure is not effective by operations other than the $puton$ operation.

Other cases can be similarly proven. □

D. Superposition of the cross-plates-unit on any type

Assume that we place the cross-plates-unit on any type of unit.

In this case, the resultant figure is always ineffective with any operation. However, the $puton$ operation succeeds for several cases.

In general, when the $puton$ operation is performed on X and Y, `WHITE` should not be placed on X's `WHITE`. When Y is a cross-plates-unit, we have to consider its white region located in the inclined orientation from $Core_Y$. The location of white region is represented as the occurrence either of b in adjacent elements or of b and w in adjacent elements in the representation for Y. For example, a representation for a unit in Figure 7 is $\langle b, b, w, g\rangle$. The sequence b, b represents the location of $white_1$, the upper left of $Core_Y$, and the sequence b, w represents that of $white_2$, the lower. Therefore, the condition on `WHITE` can be represented as (c5).

(c5) Let $\langle r_1, r_2, r_3, r_4\rangle$ and $\langle r'_1, r'_2, r'_3, r'_4\rangle$ be representations for X and Y, respectively. There exists some i ($i = 1, \ldots, 4$) such that both $r_i = r'_i \neq g$ and $r_{i+1} = r'_{i+1} \neq g$, where r_5 is regarded as r_1.

Success of extended $puton$ operation

In any pair of units X and Y, if (c1) and (c5) are satisfied, the $puton$ operation succeeds.

On the other hand, we can get valid figures by the $embed$ operation in some cases. Table I shows the result of the

fg\ bk	L1	L2	T1	T2	PLUS
I1	L1 L2	L1 L2	T1 T2	T1 T2	U*
I2	U	U	T1	T2	U
L1	L1	L2 T2	T1	T2	U*
L2	L2	L2	U*	U*	U
T1	T1	T2	T1	T2	U*
T2	U	U	T2	T2	U
PLUS	U	U	PLUS	U	PLUS

Table I
RESULT OF *embed* FOR CROSS-PLATES-UNITS

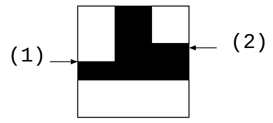


Figure 8. U*: Invalid example

embed operation. In this table, the row shows the unit in the background, and the column shows the unit in the foreground. We show the obtained types when the *embed* operation succeeds for each pair of patterns of each type. U means there is no solution. In case of U*, it appears to be successful at first glance, but there is no solution. For example, Figure 8 shows the resultant figure obtained by the operation of *embed* for L2 on T1. It is not valid because it is impossible to align line (1) and line (2).

E. Algorithm for multiple units superposition

Let Ω be a finite set of valid units where $|\Omega| \geq 2$, and ω is a W-type unit. Then, an algorithm for superposing the units in Ω is shown below.

In the following algorithm, superposition indicates either the *puton* or *embed* operation.

```
Let $\overline{\Omega}$ be a sequence obtained by setting the elements of $\Omega$ in an arbitrary order.
(1) Let $X$ be a head element $X_1$ of $\overline{\Omega}$,
    and set $\overline{\Omega} = \overline{\Omega} - \{X_1\}$.
(2) Let $Y$ be a head element $X_2$ of $\overline{\Omega}$,
    and set $\overline{\Omega} = \overline{\Omega} - \{X_2\}$.
(3) Let $Z$ be the resulting figure of superposing $Y$ on $X$.
(4) If $\overline{\Omega} = \emptyset$,
        if $Z$ is effective, then success
        else failure
    else
        if $Z$ is valid, then set $X = Z$, and goto (2)
        else failure.
```

If there exists a sequence $\overline{\Omega}$ that succeeds in this procedure, then the superposition of $\Omega \cup \{\omega\}$ succeeds.

V. CONCLUSION

We have discussed superposition of a pair of units and investigated the conditions that satisfy the result where all white regions are visible while all black regions are hidden in the resultant figure when visibility is specified by a user.

- A pair of straight-plate-units always produces an effective solution either by the *puton* operation or the *embed* operation.
- The straight-plate-unit on cross-plates-unit can produce an effective solution only by the *puton* operation.
- The cross-plates-unit on any type can produce no effective solution.

As for the last case, we have shown which pairs can generate valid solutions.

We also show an algorithm for superposing a set of units. This is the first study to address object placement with superposition.

Qualitative approach enables the reduction of computational complexity and provides intelligent reasoning by symbolic treatment of spatial data. Although it is effective on spatial database, there have been few works so far. Our contribution is to enlarge possibile application areas of qualitative spatial database by showing qualitative representation and reasoning to construct database for multiple window placement systems.

In the future, we are considering weakening the conditions of the unit, such as allowing disconnected black regions.

REFERENCES

[1] G. Birgin, R. D. Lobato, and R. Morabito, "An effective recursive partitioning approach for the packing of identical rectangles in a rectangle," *Journal of the Operational Research Society,* vol. 61, pp. 306-320, 2010.
[2] A. S. Lapaugh, "Layout algorithm for VLSI design," *ACM Computing Surveys*, vol. 28, no. 1, pp. 59-61, 1996.
[3] H. Freeman, "Computer name placement," in *Geographical Information Systems 1*, D. J. Maguire, M. F. Goodchild, and D. W. Rhind, Eds. John Wiley, 1991, pp. 449-460.
[4] J. Li, C. Plaisant, and B. Shneriderman, "Data object and label placement for information abundant visualizations," in *Proceedings of the Workshop of New Paradigms Information Visualization and Manipulation (NPIV98)*, 1998, pp. 41-48.
[5] M. Aiello, I. E. Pratt-Hartmann, and J. F. A. K.Van Benthem, Eds., *Handbook of Spatial Logics*. Springer-Verlag, 2007.
[6] A. Cohn and S. Hazarika, "Qualitative spatial representation and reasoning: an overview," *Fundamental Informaticae*, vol. 46, no. 1, pp. 1-29, 2001.
[7] M. Egenhofer and R. Franzosa, "On the equivalence of topological relations," *International Journal of Geographical Information Systems*, vol. 9, no. 2, pp. 133-152, 1995.
[8] S. Kumokawa and K. Takahashi, "Rectangle reasoning: a qualitative spatial reasoning with superposition," in *Proceedings of 23rd Florida Artificial Intelligence Research Society Conference (FLAIRS23)*, 2010, pp. 150-151.
[9] S. Wang and D. Liu, "Qualitative spatial relation database for semantic web," in *First Asian Semantic Web Conference (ASWC)*, 2006, pp. 387-399.
[10] M. Santos and L. Amaral, "Geo-spatial data mining in the analysis of a demographic database," *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, vol. 9, no. 5, pp. 374-384, 2005.