Joint Conference of 1996 International Computer Symposium
December 19~21, Kaohsiung, Taiwan, R.O.C.

Visibility Inference Based on Spatial Knowledge Representation for Environment Viewed from Observer*

P. W. Huang

Department of Applied Mathematics
National Chung-Hsing University
Taichung, Taiwan
huang@flower.amath.nchu.edu.tw

Abstract

In this paper, we propose a picture-based object-oriented spatial knowledge representation, called PCOS-string, for an environment seen from the observer's point of view. The PCOS-string representation preserves the spatial relationships among objects and thus can facilitate spatial reasoning especially from the observer's perspective. We also present a visibility inference algorithm based on the PCOS-string representation so that visibility of the objects in a given environment can be determined according to their spatial relationships. This visibility inference algorithm can handle both static and dynamic environments. Finally, we discuss the possibility of applying the PCOS-string and the associated visibility inference method to the mobile robot's path planning problem and virtual reality applications.

1. Introduction

Spatial reasoning comprises a group of methods, applied to a large group of problems, that are mainly dealing with the identification of spatial relations between objects in images or pictures [1]. For those systems that must make intelligent decisions and perform appropriate actions based on the information about an environment, spatial reasoning is extremely important to allow them to reason about the picture of the concerned environment. There are many cases in which the intelligent system must identify spatial relations that are seen from the observer's point of view. Typical examples include analysis of robot's scenes for path planning and virtual reality applications. For this type of systems, the technique that can support identification of relations corresponding to questions like "which objects are visible from a certain point" [1] is highly desired.

There are many ways for representing the concerned environment: some are descriptive and some are picture-based. In descriptive methods, the knowledge about the environment is described by a set of predicate logic statements as in STRIPS [2]. For such a method, the mapping from a real environment to a set of statements is indirect and may not truly reflect the picture of the whole environment. How the system works will heavily depend on the vocabulary chosen for describing the environment. Furthermore, it is also very tedious to use predicate logic statements to describe an environment of arbitrary complexity.

On the other hand, we may use a picture-based method to describe an environment. In picture-based methods, the knowledge about the environment is embedded in a data structure representing the picture of the environment. Many data structures have been proposed for this approach: some are pixel-oriented [3]; some utilize quadtrees [4,5], or R-trees [6,7]; and some are vector-based [8]. However, to make the system more intelligent, more flexible, and faster for real time response, the data structure should be object-oriented, and the knowledge embedded in pictures should be captured by the data structure as much as possible, especially spatial knowledge [9].

In this paper, we propose a spatial knowledge representation, called PCOS-string, for capturing the information about the picture of an environment seen from the observer's perspective. A PCOS-string can preserve the spatial relationships among the objects of a picture and thus facilitate spatial reasoning. Based on the PCOS-string representation, visibility inference for the objects in a given environment can be achieved very easily.

The remainder of this paper is organized as follows. In Section 2, we review the previous approaches to spatial knowledge representation. Section 3 describes the structure of the PCOS-string. The method of generating a PCOS-string from a given picture is also presented. In Section 4, we present a visibility inference algorithm based on PCOS-strings. The complexity of this algorithm is also analyzed. In Section 5, we discuss how to apply the PCOS-string and the visibility inference algorithm to the mobile robot's path planning problem and virtual reality applications. Finally, concluding remarks are given in the last section.

2. Previous Approaches

Considering a picture or an image map for an environment with several objects, it can be abstracted and represented as a 2D string [10] from which all spa-

*This work was supported by National Science Council, Republic of China, under Grant NSC86-2213-E-005-006

tial relationships among objects can be derived. The basic idea of the 2D string is to project the objects of a picture along the x- and y-coordinates to form two strings representing the relative positions of objects in the x- and y-axis, respectively. Since a 2D string preserves the spatial relationships among objects in a picture, it has the advantage of facilitating spatial reasoning.

Jungert [11] and Chang *et al.* [12] also extended the idea of 2D strings to form 2D G-strings by introducing several new spatial operators to represent more relative position relationships among objects of a picture. The 2D G-string representation embeds more information about spatial relationships among objects and thus facilitates spatial reasoning about shapes and relative positions of objects.

Following the same concept, Lee and Hsu [13] proposed the 2D C-string representation based on a special cutting mechanism. Since the number of parts generated by this cutting mechanism is reduced significantly, the lengths of the strings representing pictures are much shorter while still preserving the spatial relationships among objects. The 2D C-string representation is more economical in terms of storage efficiency and navigation complexity in spatial reasoning.

The Relative Coordinates Oriented Symbolic (RCOS) string proposed by Chang and Lee[14] records the symbolic names as well as the begin- and end-bounds for all objects in the picture. The spatial relationships between any two objects can be derived efficiently by a decision tree which is similar to the one proposed by Huang and Jean [15]. The length of a RCOS-string is much shorter than the length of a 2D G-string or 2D C-string as the number of objects in a picture increases [14].

The above string representations for a picture (or an image map) are all based on a Cartesian coordinates system. Basically, this type of representations is not suitable for spatial reasoning from an observer's point of view. The RS-string representation proposed by Huang and Jean [16] overcomes the above problem by using the same cutting method as the one for generating 2D C-strings except that the cutting lines are based on a polar coordinates system rather than a Cartesian coordinates system. The relation between PCOS-string (which is proposed in this paper) and RS-string is very similar to the relation between RCOS-string and 2D C-string. Thus, the PCOS-string representation is not only more economic in storage space but also more efficient in spatial reasoning when compared with the RS-string.

In the next section, we will present a spatial knowledge representation whose structure is similar to the RCOS-string except that the projections of objects are based on a polar coordinates system.

3. Polar Coordinates Symbolic String

The spatial knowledge representation discussed in this section is called the Polar Coordinates Oriented Symbolic String (or PCOS-string) whose generation is based on a polar coordinates system to maintain the view from an observer. The origin of the polar coordinates system in an image/picture plane is the position of the observer. In this image/picture plane, we can define two bounds for each object along the ring- and sector-direction, respectively. As shown in Fig. 1, the begin- and end-bounds of an object in the sector-direction are defined by the two radial segments tangent to the object while the begin- and end-bounds of an object in the ring-direction are defined by the two concentric circles tangent to the object.

Assume that the objects in the picture are enumerated from 1 to n. Let O_i be the ith object; b_i^r and e_i^r be the begin- and end-bounds of object O_i in the ring-direction; b_i^s and e_i^s be the begin- and end-bounds of object O_i in the sector-direction. Then, a PCOS-string is of the form $[O_1(b_1^r e_1^r b_1^s e_1^s), O_2(b_2^r e_2^r b_2^s e_2^s), \ldots, O_n(b_n^r e_n^r b_n^s e_n^s)]$.

A PCOS-string can be generated by the following algorithm:

Algorithm: PCOS-string Generation
Input: A picture (or segmented image) and the origin of a polar coordinates plane
Output: A PCOS-string

1. Start from the origin and scan outward along the ring-direction. The begin-bound and end-bound of an object along the ring-direction are defined by the relative distances from the origin to the two points at which the concentric circles are tangent to the boundary of the object. The point closer to the origin is called the begin-bound while the other farther point is called the end-bound of an object in the ring-direction.

2. Form the list $L_r = [O_1(b_1^r, e_1^r), O_2(b_2^r, e_2^r), \ldots, O_n(b_n^r, e_n^r)]$.

3. Draw a half line originating from the origin and place the line at the position with sector-coordinate=0 as shown in Fig. 2. From this initial position, the half line rotates clockwise and stops at the same position. When the half line rotates, the radial segments which are tangent to the boundaries of objects can be found. Assign relative coordinates to these radial segments. For every object, there are two such radial segments to form a sector. The radial segment which has a smaller sector-coordinate value is called the begin-bound while the other radial segment having a larger sector-coordinate value is called the end-bound of that object in the sector-direction.

4. If an object A happens to be cut by the rotating half line at the position with sector-coordinate=0, then A is treated as two objects A' and A'' with edge-to-edge connection at that position and the begin-bound of A' coincides with the end-bound of A'' as shown in Fig. 2. Then, A' and A'' are merged together to form the original single object A such that the begin-bound of A is the begin-bound of A'' and the end-bound of A is the end-bound of A' plus 360. If the rotating half line at the position with sector-coordinate=0 happens to be the end-bound of an object, then the coordinate of of this end-bound in the sector-direction is 360.

5. Form the list $L_s = [O_1(b_1^s, e_1^s), O_2(b_2^s, e_2^s), \ldots, O_n(b_n^s, e_n^s)]$.

6. Combine L_r with L_s to form $L=[O_1(b_1^r e_1^r b_1^s e_1^s), O_2(b_2^r e_2^r b_2^s e_2^s), \ldots, O_n(b_n^r e_n^r b_n^s e_n^s)]$. Return L.

Notice that there is no ambiguity in representing object A and object B as shown in Fig. 2, although they can be thought as having the same begin- and end-bounds in the sector-direction. Actually, the two bounds of object A and object B are represented as $(2, 361)$ and $(1, 2)$, respectively. For spatial reasoning, we can think of the position at x actually coincides with the position at $x + 360$ in the sector-direction. Furthermore, any object whose end-bound in the sector-direction is greater than 360 will be considered as an object cut by the rotating half line at the position with sector-coordinate=0.

The performance of the PCOS-string is as efficient as that of the RCOS-string because their structures are basically the same. The only difference is that the RCOS-string is based on a Cartesian coordinates system while the PCOS-string is based on a polar coordinates system. Some experimental analyses in terms of number of symbols in the string and number of operations for spatial reasoning can be found in an article by Chang and Lee [14].

4. Visibility Inference

Spatial inference for object visibility (or visibility inference for short) is the process of inferring whether an object is visible from the observer's position under some environment. There are 13 possible different spatial relationships [17] between any two non-zero sized objects in the ring- or sector-direction as shown in Fig. 3. These 13 spatial relationships can be represented by the seven spatial operators [13] whose notations and semantics are given in Table 1. What we need to do for visibility inference is to determine whether an object can be seen by the observer based on these 13 spatial relations.

In the following definitions, we assume that point P is the origin of the polar coordinates plane from which a target object is viewed. In other words, P is the position of the observer.

Definition 4.1. Object C *precedes* object B if and only if $b_C^r < b_B^r$.

Definition 4.2. A target object B is *invisible* from point P if there exists an object C preceding object B such that C = B or C] B or C % B or C [B in the sector-direction.

Definition 4.3. A target object B is *partly-visible* from point P if there exists an object C preceding object B such that B / C or C / B or B % C or B [C or B] C in the sector-direction.

Definition 4.4. A target object B is *fully-visible* from point P if it is neither *invisible* nor *partly-visible* from point P.

Before presenting the visibility inference algorithm, let us introduce the concept of *virtual blocking set* first. In a PCOS-string representation, each object is projected into the ring- and sector-direction. Therefore, there are two projection intervals associated with each object, one for the ring-direction and the other for the sector-direction. A projection interval is represented as a pair (b e), where b is the beginning point and e is the ending point of that interval. Assume that object O_1 has the projection interval (b_1^s e_1^s) and object O_2 has the projection interval (b_2^s e_2^s) in the sector-direction. We say that the two intervals can be **merged** in the sector-direction if and only if the intersection of the above two projection intervals is not empty. Then, the interval ($min\{b_1^s, b_2^s\}$ $max\{e_1^s, e_2^s\}$) is called the *connected projection interval* merged from object O_1 and object O_2 in the sector-direction. A *maximal connected projection interval* in a given direction is an interval such that no other projection intervals in the same direction can be **merged** with it. The *virtual blocking set* for a given object O_T is the set of all maximal connected projection intervals in the sector-direction that merged from the projection intervals along the sector-direction for all objects preceding object O_T. Let us look at the example shown in Fig. 4, the virtual blocking set for object O_T is the set of intervals {(1, 5), (7, 8), (9, 10)}.

Without loss of generality, we assume that there are n objects O_1, O_2, O_3, ..., O_n in a PCOS-string such that $b_{i-1}^r \le b_i^r$ for $2 \le i \le n$. Let V_i be the visibility of object O_i viewed from the origin and B_i be the virtual blocking set for object O_i. Note that it is meaningful to compare an interval I with the projection of an object O using one of the seven spatial operators because they have the same representation. We also use the notation $P_s(O)$ to indicate the projection of object O along the sector-direction. Now we are ready to present the visibility inference algorithm as follows.

Algorithm: Spatial inference for object visibility
Input: A PCOS-string: $[O_1(b_1^r e_1^r b_1^s e_1^s), O_2(b_2^r e_2^r b_2^s e_2^s), \ldots, O_n(b_n^r e_n^r b_n^s e_n^s)]$ with $b_1^r \le b_2^r \le \ldots \le b_n^r$.
Output: The list $\{V_1, V_2, V_3, \ldots, V_n\}$.

1. $i \leftarrow 0$; $M \leftarrow \emptyset$. /* Initialization */

2. Find the next m objects $\{O_{i+1}\ O_{i+2}\ \ldots\ O_{i+m}\}$ with $b_{i+1}^r = b_{i+2}^r = \ldots = b_{i+m}^r$.

3. For $j = 1$ to m do
   $B_{i+j} \leftarrow M$ and

   (a) If there exists an interval $I \in B_i$ such that
   $I = P_s(O_{i+j})$ or
   I [$P_s(O_{i+j})$ or
   I] $P_s(O_{i+j})$ or
   I % $P_s(O_{i+j})$,
   then $V_{i+j} \leftarrow$ ***invisible*** else

   (b) If there exists an interval $I \in B_i$ such that
   I / $P_s(O_{i+j})$ or
   $P_s(O_{i+j})$ / I or
   $P_s(O_{i+j})$ [I or
   $P_s(O_{i+j})$] I or
   $P_s(O_{i+j})$ % I,
   then $V_{i+j} \leftarrow$ ***partly-visible*** else

   (c) $V_{i+j} \leftarrow$ *fully-visible.*

4. If $i+m=n$ then return $\{V_1\ V_2\ \ldots\ V_n\}$ else

   (a) From $O_{i+1}, O_{i+2}, \ldots, O_{i+m}$ and B_i, compute the *virtual blocking set* M for object O_{i+m+1}.

   (b) $i \leftarrow i+m$; GoTo 2.

Assume that the major cost is in the comparison operations. Preparing a valid PCOS-string as the input for the above visibility inference algorithm requires sorting on a list of O_i's by b^r for $1 \le i \le n$. Applying a good sorting algorithm will take time $O(n \log n)$. Time required by step 1 is $O(1)$. Total time spent on step 2 is $O(n)$. Total time spent on step 3 is $O(n^2)$ because there are n objects in the PCOS-string and there might be $i-1$ intervals that need to be inspected in order to determine the visibility of object O_i in the worst case and each inspection requires a finite number of comparisons to determine the spatial relationship between two concerned intervals. Total time spent on step 4 to compute the virtual blocking sets for all n objects is $O(n^2)$ in the worst case. Thus, the overall time complexity of the above algorithm is still $O(n^2)$.

There are two situations in which the visibility of an object may be affected: (1) changing the observer's position; (2) objects are removed or added under a dynamic environment. In the former case, we need to regenerate a new PCOS-string to represent the environment using the observer's new position as the origin of the underlying polar coordinates system. Then, we re-calculate the visibility of every object by the above inferencing algorithm. In the latter case, we only re-compute the visibility for some objects. If O_i is the object deleted from the environment, the visibility of all objects O_j with $b^r_j \le b^r_i$ should not be affected. Thus, we only need to re-compute the visibility and the virtual blocking sets for the objects O_k with $b^r_k > b^r_i$. If O_a is the object added into the environment such that $b^r_i \le b^r_a < b^r_{i+1}$, then we re-compute the visibility and the virtual blocking sets for object O_a and all objects O_k with $i+1 \le k \le n$. By choosing an appropriate starting object as described above, the same inference algorithm can be used to update the visibility and the virtual blocking sets for the affected objects. Thus, our visibility inference algorithm is suitable for both static and dynamic environments.

5. Possible Applications

Visibility inference based on the PCOS-string representation can be applied to the path planning problem of a mobile robot navigation system and virtual reality applications. In a mobile robot navigation system, the autonomous robot should be intelligent enough to map perceptions onto appropriate actions so that it can travel through the concerned environment while avoiding obstacles. One of the major functions of such an intelligent system is to search for a continuous set of points connecting the initial position of the robot to its desired destination. If the robot is so small as to be considered a point, the problem can be solved straightforwardly by constructing a visibility graph [18]. Let S be the set consisting of the initial and final positions as well as the vertices of all obstacles. To form the visibility graph, we connect every pair of points in S that are visible from one another. This step can be achieved by our visibility inference algorithm provided that we have a PCOS-string representing the environment viewed from the robot. However, such a string can be generated by our PCOS-string generation algorithm from a pre-drawn picture for the environment or an image map taken by the overhead camera. Then, we can search the visibility graph to find an optimal path for the robot perhaps by using the A* algorithm [19].

In virtual reality applications, the observer is surrounded by many objects among which some are fully visible, some are partly visible, and the rest are totally invisible. The system will simulate the scene viewed from the observer. From time to time, objects may be removed from or added into the simulated environment and the observer's view should be updated accordingly. Since the simulated environment can be represented by a PCOS-string, the visibility for all objects can be easily determined by our visibility inference algorithm. The visibility of objects can be easily adapted to the changing environment by using the same visibility inference algorithm with a minor modification as we have described in Section IV.

6. Conclusions

Traditionally, there are descriptive methods and picture-based methods to represent a physical environment under consideration. A descriptive method relies on a set of predicate logic statements to make reasoning possible. Such an approach may not truly reflect the whole picture of a complex environment. On the other hand, although the pixel-oriented method, quadtrees, R-trees, and the vector-based method are all picture-based, they are not object-oriented, therefore, difficult and inefficient for spatial reasoning.

In this paper, we propose a spatial knowledge representation, called PCOS-string, for capturing the information about the picture of an environment seen from the observer's perspective. Since a PCOS-string is object-oriented and can preserve the spatial relationships among the objects in a picture, this type of representation can facilitate spatial reasoning in a very efficient way. We also propose a visibility inference algorithm based on the PCOS-string representation. This inference algorithm is suitable for both static and dynamic environments. Finally, we discuss the possibility of applying the PCOS-string and the associated visibility inference mechanism to the path planning problem in a mobile robot navigation system and virtual reality applications.

References

[1] E. Jungert, "The observer's point of view: an extension of symbolic projections," In *International Conference on Methods for Spatial and Tempo-*

ral Reasoning in GIS. Pisa, Italy, 179-195 (Sept 1992).

[2] R.E. Fikes and N.J. Nilsson, "STRIPS: A new approach to the application of theorem proving to problem solving," *Artificial Intelligence* **2**, 189-208 (1971).

[3] M. Chock, A.F. Cardenas, and A. Klinger, "Database structure and manipulation capabilities of the picture database management system (PICDMS)," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **PAMI-6**, 484-492 (1984).

[4] H. Samet, "The quadtree and related data structures," *ACM Computing Surveys*, vol. 16, no. 2, 187-260 (1984).

[5] Subbarao Kambhampati and Larry S. Davis, "Multiresolution path planning for mobile robots," *IEEE Journal of Robotics and Automation*, **RA-2**, no. 3, 135-145 (1986).

[6] A. Guttman, "R-trees: A dynamic index structure for spatial searching," in *Proc. ACM-SIGMOD Int. Conf. Management of Data*, 47-57 (1984).

[7] N. Roussopoulos and D. Leifker, "Direct spatial search on pictorial databases using packed R-trees," in *Proc. ACM-SIGMOD Int. Conf. Management of Data*, 17-31 (1985).

[8] E. Jungert *et al.*, "Vega - A geographical information system," in *Proc. Scandinavian Res. Conf. Geographical Information Systems*, Linkoping, Sweden (June 1985).

[9] S.K. Chang, C.W. Yan, Donald C. Dimitroff, and Timothy Arndt, "An intelligent image database system," *IEEE Trans. on Software Engineering* **14**, 681-688 (May 1988).

[10] S.K. Chang, Q.Y. Shi, and C.W. Yan, "Iconic indexing by 2-D strings," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **PAMI-9** ,413-428 (May 1987).

[11] E. Jungert, "Extended symbolic projections as a knowledge structure for spatial reasoning," *4th BPRA Conf. on Pattern Recognition*, Springer, Cambridge, 343-351 (March 1988).

[12] S.K. Chang, E. Jungert, and Y. Li, "Representation and retrieval of symbolic pictures using generalized 2D strings," *Technical Report*, University of Pittsburg (1988).

[13] S.Y. Lee and F.J. Hsu, "2D C-string: a new spatial knowledge representation for image database system," *Pattern Recognition* **23**, 1077-1087 (1990).

[14] Chin-Chen Chang and Chin-Feng Lee, "Relative coordinates oriented symbolic string for spatial relationship retrieval," *Pattern Recognition* **28**, 563-570 (1995).

[15] P.W. Huang and Y.R. Jean, "Using 2D C^+-string as spatial knowledge representation for image database systems," *Pattern Recognition* **27**, 1249-1257 (1994).

[16] P.W. Huang and Y.R. Jean, "Spatial reasoning and similarity retrieval for image database systems based on RS-strings," to appear in *Pattern Recognition* (1996).

[17] J.F. Allen,"Maintaining knowledge about temporal intervals," *Communications of the ACM* 26(11), 832-843 (1983).

[18] Elaine Rich and Kevin Knight, *Artificial Intelligence*, 2nd Edition, McGraw-Hill, Inc., New York, 570-571 (1991).

[19] N.J. Nilsson, *Principles of Artificial Intelligence*, Tioga, Palo Alto, CA. (1980).

Table 1. The definitions of spatial operators.

Notations	Conditions
$A < B$	$end(A) < begin(B)$
$A = B$	$begin(A) = begin(B), end(A) = end(B)$
$A\|B$	$end(A) = begin(B)$
$A\%B$	$begin(A) < begin(B), end(A) > end(B)$
A[B	$begin(A) = begin(B), end(A) > end(B)$
A]B	$begin(A) < begin(B), end(A) = end(B)$
A/B	$begin(A) < begin(B) < end(A) < end(B)$

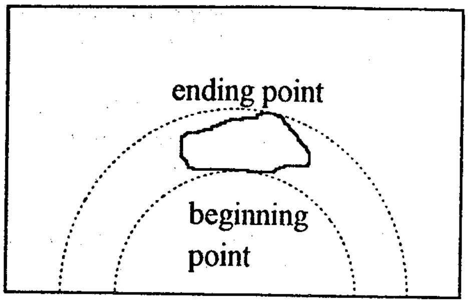

Fig. 1(a). The begin- and end-bounds of an object in the ring-direction.

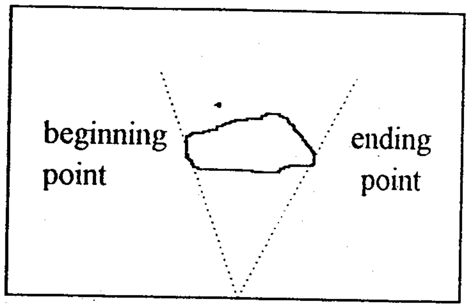

Fig. 1(b). The begin- and end-bounds of an object in the sector-direction.

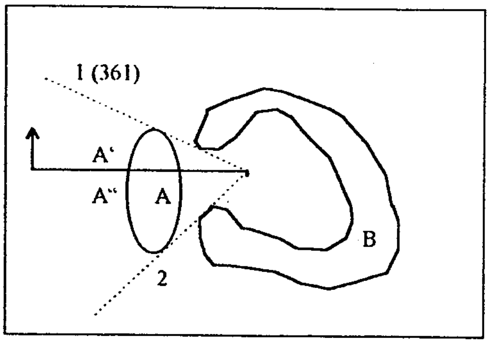

Fig. 2. The initial position of the rotating half line is at sector-coordinate=0. Object A is cut into subparts A' and A'' by the half line at this position. The two bounds of object A and object B are represented as (2, 361) and (1, 2), respectively.

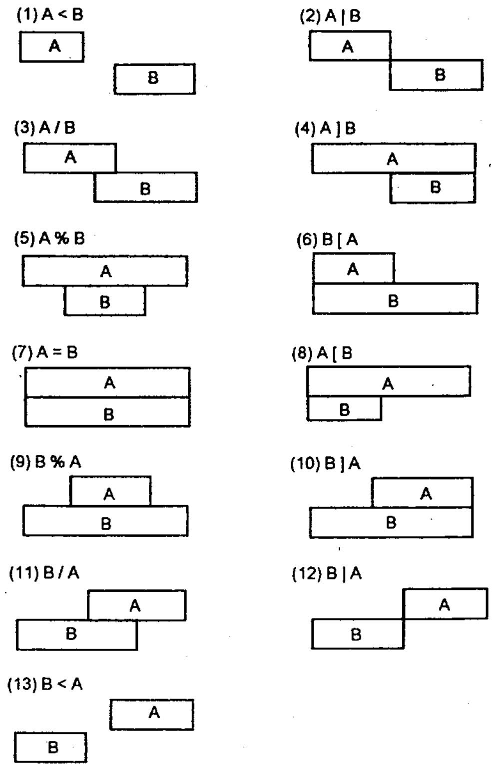

Fig. 3. The 13 possible different spatial relationships between any two objects in one direction.

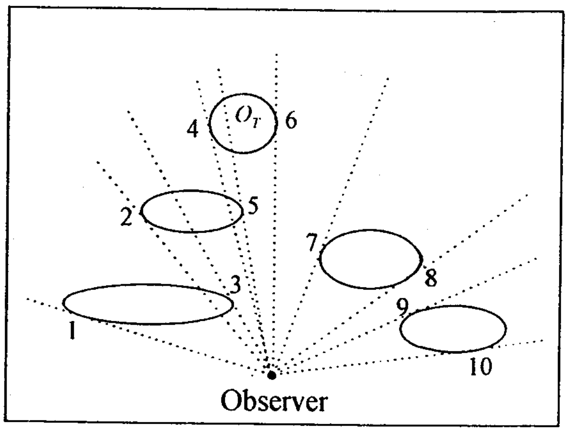

Fig. 4. Assume that all objects precede O_T and the numbers indicate the begin- and end-bounds of all all objects in the sector-direction. The virtual blocking set for object O_T is $\{(1,5),(7,8),(9,10)\}$.