

PII: S0031-3203(96)00186-0

A PRIME-NUMBER-BASED MATRIX STRATEGY FOR EFFICIENT ICONIC INDEXING OF SYMBOLIC PICTURES¹

YE-IN CHANG* and BI-YEN YANG

Department of Applied Mathematics, National Sun Yat-Sen University, Kaohsiung, Taiwan, R.O.C.

(Received 6 February 1996; in revised form 21 November 1996)

Abstract—In the previous approaches to represent pictorial data, as the complexity of the representation strategy is increased, the more spatial relationships can be represented, which also results in a more complex strategy for query processing and a limited types of queries that can be answered. In this paper, we propose an efficient iconic indexing scheme called *Prime-Number-based Matrix* (PN Matrix) for symbolic pictures, which combines the advantages of the 2D *C-string* and the *9DLT matrix*. Basically, the proposed strategy can represent those complex relationships which are represented in 2D C-strings in a matrix, and an efficient module-based operation can be used to support pictorial query, spatial reasoning and similarity retrieval. In the proposed scheme, we classify 169 spatial relationships between two objects in 2D space into five spatial categories, and define a *category rule* for each of those five spatial categories. Those category rules are module-operation-based; therefore, they are efficient enough as compared to the previous approaches. Following those category rules, we propose algorithms to efficiently support spatial reasoning, picture queries and similarity retrieval based on a data structure of a Prime-Number-based Matrix (PN Matrix). © 1997 Pattern Recognition Society. Published by Elsevier Science Ltd.

2D string 2D C-string Image databases Pictorial query Similarity retrieval Spatial reasoning Symbolic databases

1. INTRODUCTION

The design of image databases has attracted much attention over the past few years. Applications which use image databases include office automation, computer aided design, robotics, and medical pictorial archiving. A common requirement of these systems is to model and access pictorial data with ease.⁽¹⁾ Thus, one of the most important problems in the design of image database systems is how images are stored in the image database.⁽²⁾ In traditional database systems, the use of indexing to allow database accessing has been well established. Analogously, picture indexing techniques are needed to make ease pictorial information retrieval from a pictorial database.⁽³⁾

Over the last decade, many approaches to represent symbol data have been proposed, as shown in Fig. 1. Chang *et al.*⁽²⁾ propose a pictorial data structure, 2D *string*, using symbolic projections to represent symbolic pictures preserving spatial relationships among objects.⁽⁴⁾ The basic idea is to project the objects of a picture along the x- and y-axis to form two strings representing the relative positions of objects in the xand y-axis, respectively.⁽⁵⁾ A picture query can also be specified as a 2D string. Based on 2D strings, several algorithms in pictorial querying, spatial reasoning and similarity retrieval are proposed, where picture querying allows the users to query images with a specified spatial relationship, spatial reasoning means the inference of a consistent set of spatial relationships among the objects in an image, and the target of similar retrieval is to retrieve the images that are similar to the query image. Based on the 2D string longest common subsequence strategy, Lee et al.⁽⁶⁾ propose an algorithm for similarity retrieval. Based on an object's and spatial relationship (OSR) table strategy, Liaw⁽⁷⁾ also proposes an algorithm for similarity retrieval, which distinguishes whether two images are similar by comparing two sets of real numbers that are associated to these images, respectively. Besides, Costagliola et al.⁽³⁾ introduce a variation of 2D string representation for symbolic pictures, the non-redundant 2D string, which is a more compact representation than the 2D string.

However, the representation of 2D strings is not sufficient enough to describe pictures of arbitrary complexity completely. For this reason, Jungert^(8,9) and Chang *et al.*⁽¹⁰⁾ introduce more spatial operators to handle more types of spatial relationships among objects in image databases. Using these extended spatial operators, 2D *Gstring* representation facilitates spatial reasoning about shapes and relative positions of objects. But a 2D Gstring representation scheme is not ideally economic for complex images in terms of storage space efficiency and navigation complexity in spatial reasoning. Therefore, Lee and Hsu⁽⁴⁾ propose a 2D *C*-string representation scheme. Since the number of subparts generated by this

^{*} Author to whom correspondence should be addressed. Tel: 887-7-5252000 (ext. 3819); fax: 886-7-5253809; e-mail: changyi@math.nsysu.edu.tw.

¹This research was supported in part by the National Science Council of Republic of China under Grant No. NSC-84-2213-E-110-009.



Fig. 1. Approaches to symbolic picture representation.

new cutting mechanism is reduced significantly, the lengths of the strings representing pictures are much shorter while still preserving the spatial relationships among objects. However, the previous abstractions ignore the relative sizes and locations of objects. Thus, similarity retrieval of images may be ambiguous and many types of queries concerning sizes, locations, and distances cannot be answered due to this inadequacy. Therefore, Huang and Jean⁽⁵⁾ propose a 2D C^+ -string representation scheme which extends the work of Lee and Hsu,⁽⁴⁾ 2D C-string, by including relative metric information about the picture into the strings.

As described before, based on 2D string representation, the problem of picture query turns out to be the matching of 2D subsequences, which takes non-polynomial time complexity. This makes the picture retrieval method inappropriate for implementation, especially when the number of objects in an image is large. Therefore, Chang et al.⁽¹¹⁾ propose an efficient approach of iconic indexing by a nine direction lower-triangular (9DLT) matrix. Moreover, to handle large amounts of image databases, a simple algorithm for spatial match retrieval of symbolic pictures based upon 9DLT matrices is proposed by Chang et al.⁽¹²⁾ by using the concept of superimposed coding, which reduces the ratio of false match when a query picture occurs. Next, Chang et al.⁽¹³⁾ propose a module-oriented signature extraction strategy, in which prime numbers are used to compose the signatures, and the module operation will be applied when the query happens.

In the previous approaches to represent pictorial data, as the complexity of the representation strategy is increased, the more spatial relationships can be represented, which also results in a more complex strategy for query processing and a limited types of queries that can be answered. In this paper, we propose an efficient iconic indexing scheme called *Prime-Number-based Matrix* (PN Matrix) for symbolic pictures, which combines the advantages of the 2D C-string and the 9DLT *matrix*. Basically, the proposed strategy can represent those complex spatial relationships that are represented in 2D C-strings in a matrix, while it does not need any cutting strategy and complex procedures to do spatial reasoning. Moreover, the proposed strategy can be considered as an extended 9DLT matrix strategy in which more than nine spatial relationships can be represented and an efficient module-based operation can be used to support pictorial query, spatial reasoning and similarity retrieval. In the proposed scheme, we classify those 169 spatial relationships between two objects in 2D space as observed by Lee and Hsu⁽⁴⁾ into five spatial categories, and define a category rule for each of those five spatial categories. Those category rules are module-operationbased; therefore, they are efficient enough as compared to the previous approaches. Following those category rules, we propose algorithms to efficiently support spatial reasoning, picture queries and similarity retrieval based on a Prime-Number-based Matrix (PN Matrix) strategy. In a PN Matrix, the relationships between two objects along the x-axis (or y-axis) is recorded in a number which is a product of some prime numbers. Therefore, spatial reasoning can be done very straightforwardly. For answering a pictorial query, some module operations or membership checking of a set of numbers are applied. In similarity retrieval, some new matrix operations are applied.

The rest of the paper is organized as follows. In Section 2, we give a brief description about the previous symbolic picture representations. In Section 3, we will present the proposed efficient iconic indexing scheme for symbolic pictures. Finally, Section 4 gives a conclusion.

2. BACKGROUND

In this section, we briefly describe several data structures for symbolic picture representation, including 2D string, 2D G-string, 2D C-string and 9DLT matrix. Moreover, we will briefly describe the strategy for spatial reasoning based especially on the 2D C-string representation.

2.1. 2D string

For pictorial information retrieval, Chang *et al.*⁽²⁾ present a new way of representing a picture by a 2D string, and a picture query can also be specified as a 2D

d		
	b	с
a	e f	

Fig. 2. A picture f.

string. The problem of pictorial information retrieval then becomes a problem of 2D subsequence matching.

Let V be a set of symbols, where each symbol could represent a pictorial object or a pixel. Let A be the set $\{=, <, :\}$, where "=", "<" and ":" are three special symbols not in V. For example, consider the picture shown in Fig. 2, $V = \{a, b, c, d, e, f\}$. The 2D string representing the above picture f is as follows:

$$(a = d < e : f = b < c, a = e : f < b = c < d),$$

where the symbol "<" denotes the left-right or belowabove spatial relationship. The symbol "=" denotes the "at the same spatial location as" relationship and the symbol ":" denotes the "in the same set as" relation.

2.2. 2D G-string

The spatial operators ">" and "=" are not sufficient enough to give a complete description of spatial knowledge for pictures of arbitrary complexity. To represent the spatial relationship between two non-zero sized objects, especially for the case of overlapping objects, Jungert⁽⁸⁾ introduces some local operators as compensation for handling more types of relationships between pictorial objects in query reasoning. Later, Chang *et al.*⁽¹⁰⁾ introduce the *generalized 2D string* (*2D G-string*) with a cutting mechanism.

Jungert extends the operators of 2D strings as global operators R_g and introduces a set of local operators R_l to handle more types of spatial relationships among non-zero sided objects (i.e. the partial overlapping relationships), which are defined as follows, where the definitions of those spatial operators are given in Table 1:

$$R_{g} = \{<, =, |\}, \quad R_{l} = \{/, \%, [,]\}.$$

The cuttings are performed at all extreme points of all the objects to segment the objects in the image. One example is shown in Fig. 3, and the corresponding 2D G-string representation is as follows:

$$2D G_{-}x - \operatorname{string}(f) :$$

$$A|A = B|A = B = D|A = D|D|C = D|D|$$

$$2D G_{-}y - \operatorname{string}(f) :$$

$$D < B|B = C|A = B = C|A = C|A.$$

2.3. 2D C-string

In 2D G-string representation, the number of segmented subparts of an object is dependent of the number of bounding lines of other objects which are completely or

Table 1. Definitions of Jungert's spatial operators

Notation	Condition	Meaning
A < B	$\operatorname{center}(A) < \operatorname{center}(B)$	
A = B	$\operatorname{center}(A) = \operatorname{center}(B)$	A B
A B	edge to edge with	A
A%B	min(A) > min(B) max(A) < max(B) length(A) < length(B)	
A[B	$\min(A) = \min(B)$ length(A) < length(B)	
<i>A</i>] <i>B</i>	$\max(A) = \max(B)$ length(A) < length(B)	A
A∖B	$\min(A) < \min(B)$ length(A) \leq length(B)	
A/B	$\max(A) > \max(B)$ length $(A) \le \text{length}(B)$	



Fig. 3. The cutting mechanism of the 2D G-string: (a) cut along the x-axis; (b) cut along the y-axis.

partly overlapping with this targeted object. For the cases of objects with overlapping, the storage space overhead is high and it is time consuming in spatial reasoning. Therefore, to overcome this problem, a 2D C-string is proposed by Lee and Hsu.⁽⁴⁾ A more efficient and economic cutting mechanism is described by employing a sound and characteristic set of spatial operators.

2.3.1. Representation. Table 2 shows the formal definition of the set of spatial operators, where the notation "begin(A)" denotes the value of begin-bound of object A and "end(A)" denotes the value of end-bound of object A. According to the begin-bound and end-bound of the picture objects, spatial relationships between two enclosing rectangles along the x-axis (or y-axis) can be categorized into 13 types ignoring their length. Therefore, There are 169 types of spatial relationships between two rectangles in 2D space, as shown in Fig. 4. Basically, a cutting of the 2D C-string is performed at the point of partly overlapping, and it

Table 2. Definitions of Lee's spatial operators

Notation	Condition	Meaning
$\overline{A < B}$	end(A) < begin(B)	A disjoins B
A = B	begin(A) = begin(B) end(A) = end(B)	A is the same as B
A B	end(A) = begin(B)	A is edge to edge with B
A%B	$\operatorname{begin}(A) < \operatorname{begin}(B)$	A contains B and they
	$\operatorname{end}(A) > \operatorname{end}(B)$	do not have the same bound
A[B]	begin(A) = begin(B)	A contains B and they
	$\operatorname{end}(A) > \operatorname{end}(B)$	have the same begin bound
A]B	$\operatorname{begin}(A) < \operatorname{begin}(B)$	A contains B and they
	$\operatorname{end}(A) = \operatorname{end}(B)$	have the same end bound
A/B	$\operatorname{begin}(A) < \operatorname{begin}(B)$	A is partly overlapping
	end(A) < end(B)	with B

Disj	joint (4	48)	J	oint (4	40)	Part	_ovlp	(50)	Con tain (16)	Bel ong (16)
_ Ø	_ <u>222</u> _ < % <	⊠_* I'			 ⊒_1][*	⊏ ¶] %*]]*		
□ <mark>⊠</mark> <		₿□ <* /'		昭 /*1	₩ % *	-∰ %['	₩ %%*	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	=]	=]*
	=	Ø□ <*]	[] []	*		2 [[*	8 [%*	E]*	= %	=%*
□⊠ <]	[* <	⊠⊡ <*%*	□¤ 1%	22223 %*	= I*	222]* [ezz a]* %]*]	= [= [*
[@	₩ 2223 0 %*<	Ø□ <*[□ 2224 1[[*	/* I*	∞ %*[₽ ₽ ₽ %*%	22 6 23 %*]]=	2]*=
□⊠ <[]* <	□ 1 <*	[][[]]	₽ */	2222]*I*	[* [[*%	[*]	[]	2222]*]*
□⊠ < =	/* <		□ ■ ■	₽ [*]	%*I*	⊡≣ 20 //	1	[⊉ 223 ∕%]%	255]*%*
□ 	⊠ * <]<*	□ 1/*	₽22 *%	[* *] 	□ ■ 22 / =	1 //*][222]*[*
□ ∞ * *	⊠ <* <	□ % <*	□ ⊠]*	₽222 * [□ 12 2	⊐12 /%*	-₩ 3 / [*	₩ % =	2 2 2 %*=
□ < /*	<u>_</u> 18	,_∎ ,	<u>-</u> []	"*]/	\$ B	7	8	<u>[</u> %*]*
□ Ø <]*	□ <u>⊠</u> <*/	, ,	□Ø [*			= /	Ø /*/]*/	* %	** %*
□Ø <%*	⊠□ <*]	, ⊾	/1	₩,		%*/	[*/	□ 22 /*]	% [%*[*
□Ø <[*	⊠∏ <* %	¥∎][Ø *]*		⊠∎ ∕*%	22 /* [Ø∎ /*≈	[=	⊠ [*=
□ <	⊠□ <*[]*<*	- %1	⊠⊐ *%*		Ø - /* /*	/*]*	Ø /*%*	[]	[*]*
 / <	Ø□ <* =	,⊒⊒ %*<*	<u>₽</u> [Ø⊟ * [*	[* /*	Ø=] /*	₩ %/*	[%	₩Ø [*%*
]<	□ ≈*<*	□ [*<*	= 1	7 / *	2	2 [/*	= /*	2213]* /*	[[[* [*

Fig. 4. The 169 spatial relationship types of two objects.

keeps the former object intact and partitions the latter object. The cutting mechanism is also suitable for pictures with many objects. Furthermore, the end-bound point of the dominating object does not partition other objects which contain the dominating object. Less cuttings and no unnecessary cuttings in 2D-C string will make the representation more efficient in the case of overlapping as shown in Fig. 5. The corresponding 2D C-string is as follows:

$$2DC_x - string(f) : A]B]C|A = D|D\%C,$$

$$2DC_y - string(f) : D < B]C|A|A[C.$$



Fig. 5. The cutting mechanism of the 2D C-string: (a) cut along the x-axis; (b) cut along the y-axis.

2.3.2. Spatial reasoning. To solve the problem of how to infer the spatial relations along the x-axis (or yaxis) between two pictorial objects from a given 2D Cstring representation, the *level* and *rank* of a symbol are used.⁽⁴⁾ The rank of each symbol in a 2D string proposed by Chang *et al.*⁽²⁾ is defined to be one plus the number of "<" preceding this symbol in a x-string or y-string. That is, the rank values of objects stand for the relative sequence in the x-string or y-string representing the relative spatial position in the original symbolic picture. Because the 2D C-string representation of symbolic pictures is constructed by employing more spatial operators, R_g and R_l , the ranks of pictorial objects need to be redefined.⁽⁴⁾

Suppose s_i is a symbol of x(y) string, the rank of s_i is denoted as $\operatorname{Rank}(s_i) = k_1^i, k_2^i, \ldots, k_{l_i}^i$, where l_i is the rank level of s_i , denoted as $\operatorname{Level}(s_i) = l_i, k_m^i$ is the rank value of the *m*th level of $s_i, m \in [1, l_i]$. The ranking technique is actually an encoding method. The level l_i of symbol s_i means the depth of nesting within a complex object or a local body, which is used in the following rank rules. Following nine complex rank rules recursively, the rank of any symbol in a 2D C-string can be defined.⁽⁴⁾ For instance, the 2D C-string along the x-axis of Fig. 6 is as follows:

$$A|B[(C < D\%(E = F](G]H < J) < K[L) < M,$$

where "()" is a pair of separators which is used to describe a set of symbols as one local body.

(



Fig. 6. An example of 2D C-string representation.

Then, the ranks and levels of these symbols along the x-axis are as follows:⁽¹⁴⁾

Rank(A) = 1	Level(A) = 1
$\operatorname{Rank}(B) = 2$	Level(B)=1
$\operatorname{Rank}(C) = 2.0$	Level(C)=2
Rank(D)=2.2	Level(D)=2
$\operatorname{Rank}(E) = 2.2.1$	Level(E)=3
$\operatorname{Rank}(F) = 2.2.1$	Level(F)=3
Rank(G) = 2.2.1.1	Level(G)=4
Rank(H)=2.2.1.1.101	Level(H) = 5
Rank(I)=2.2.1.103	Level(I)=4
Rank(J)=2.2.102	Level(J)=3
Rank(<i>K</i>)=2.4	Level(K)=2
$\operatorname{Rank}(L) = 2.4.0$	Level(L)=3
$\operatorname{Rank}(M) = 4$	Level(M) = 1

The spatial knowledge is embedded in the ranks of pictorial objects. In fact, the ranks become representative of the spatial knowledge of pictorial objects in an image. All the spatial relationships except "/" can be inferred by using the ranks. Because the cutting mechanism of a 2D C-string is performed for the case of "/", it is impossible for any two segmented subobjects or symbols to be partly overlapping.⁽¹⁴⁾ To identify the spatial relationships along the x-axis (or y-axis) between two symbols using their ranks, six complex computing rules are used.⁽⁴⁾ For example, assume that there are two symbols s_i , s_i in a picture, $\operatorname{Rank}(s_i) = \operatorname{rank}_i = k_1^i \cdot k_2^i \cdots k_{l_i}^i$, $\operatorname{Level}(s_i) = l_i$, $\operatorname{Rank}(s_j) = \operatorname{rank}_j = k_1^j \cdot k_2^j \cdots k_{l_j}^j$, $\operatorname{Level}(s_j) = l_j$, and the rank values of the first (h-1) levels of s_i are equal to those of s_j , i.e. $k_1^i = k_1^j, k_2^i = k_2^j, \dots, k_{(h-1)}^i = k_{(h-1)}^j$, and $k_h^i \neq k_h^j$. In this case, the following computing rule determines whether a symbol s_i is edge to edge with a symbol s_j or not. If $\exists h \in [1, \min(l_i, l_j)]$, $mod(k_h^j - k_h^i, 100) = 1$, and $\forall x \in [h+1, l_i], k_x^i > 100$, and $\forall y \in [h+1, l_j], k_y^j = 0$, then $s_i | s_j$.

Furthermore, to answer the spatial relationship between two objects along the x-axis (or y-axis), which are segmented into subparts, we have to compare all subparts of the objects. In general, according to the spatial relationship between these two objects' boundary subparts, there are four cases possible. For each case, up to two comparisons between the leftmost (or rightmost) bounding subpart of those two objects are needed to determine the spatial relationship.⁽¹⁴⁾

2.3.3. Pictorial query. The primitive direction relationships can be inferred from the spatial operators

of 2D C-strings. Four basic orthogonal directional aggregates are the main body of a 2D C-query. For example, to determine whether A is in the east of B, the following aggregate is used: $x : Ar_{A,B}^x B, r_{A,B}^x \in \{<^*, |^*, [, \%, /^*\}$ iff (east, A, B).⁽¹⁴⁾ In general, the pictorial queries in a 2D C-query can be summarized and classified into seven classes. For example, in the category of a relationship object query, to determine what objects overlap with X, the following rule is used:

$$partovlp, ?, X)$$

$$= \{A | ((x_partovlp, A, X) - (y_disjoin, A, X) - (y_disjoin, A, X) - (y_edge, A, X)) \cup ((y_partovlp, A, X) - (x_disjoin, A, X) - (x_edge, A, X)) \cup ((x_more, A, X) \cap (y_less, A, X)) \cup ((x_eless, A, X) \cap (y_more, A, X)) \},$$

where x_{-} partovlp (y_{-} disjoin) is one of 12 1D relationship aggregates along the x-axis (y-axis). (Note that spatial reasoning is the basic work to the pictorial query. That is, the rank rules and *computing* rules are applied in a pictorial query, too.)

2.3.4. Similarity retrieval. The target of similarity retrieval is to retrieve the images that are similar to the query image. Based on the maximum-likelihood or minimum-distance criterion, a new definition of type-*i* similar pictures in 2D C-strings is proposed by Lee and Hsu.⁽¹⁴⁾ Basically, similarity retrieval in 2D C-string representation is processed as follows. First, the representing 2D C-strings for the two pictures f_1 and f_2 are constructed. By applying the reasoning rules, the spatial relationships and categories among objects are inferred. According to the definition of type-*i* similar picture, the set of type-*i* similar pairs of picture f_1 and f_2 is constructed. Finally, the maximal complete subgraphs of type-0, type-1 and type-2 are found.

2.4. 9DLT matrix

Chang et al.⁽¹¹⁾ classify spatial relationship into nine classes, according to the x-axis and y-axis spatial relative information embedded in the picture, and suggest a nine direction lower-triangular (9DLT) matrix to represent a symbolic picture. Let there be nine direction codes (as shown in Fig. 7) which are used to represent relative spatial relationships among objects. In Fig. 7, R denotes the referenced object, 0 represents "at the same spatial location as R", 1 represents "north of R", 2 represents "northwest of R", 3 represents "west of R" and so on. The symbolic picture shown in Fig. 8(a), Fig. 8(b) is the corresponding 9DLT matrix.

3. AN EFFICIENT ICONIC INDEXING SCHEME FOR SYMBOLIC PICTURES

In general, Lee and Hsu's $algorithm^{(14)}$ for spatial reasoning based on 2D C-strings can be summarized into the following three steps: (1) Following rank rules



Fig. 7. The direction codes.



Fig. 8. 9DLT representation: (a) a symbolic picture; (b) the related 9DLT matrix.

recursively, the rank value of each symbol is calculated. (2) Following computing rules, the spatial relationships between two symbols are inferred. (3) To infer the spatial relationship between two partitioned objects, the boundary of their subparts is compared. Consequently, to answer a pictorial query based on 2D C-string representation, a lot of steps are needed. Therefore, in this section, we propose a new iconic indexing scheme which can solve spatial queries easier and more efficiently. By observing the 169 spatial relationships in Fig. 4, we can classify them into five spatial categories: disjoin, join, contain, belong and part_overlap, as shown in Figs 9-13, respectively. According to these five figures of these spatial categories, we discover some interesting and useful results. A category rule is derived for each spatial category. Following those category rules, we propose algorithms to efficiently support spatial reasoning, picture queries and similarity retrieval based on

_ Ø	222 %<	⊠_*I*	[]ஜ	₩ 20%*<	⊠⊡ <*[*	□ < *	⊠ ~*<	[ø %<*	□Ø <[*	⊠⊒[] <*%	/*<*
□ <mark>∅</mark> <	⊠[<	Ø□ <*/*	□ <[⊠222 □]*<	□ 2223 <*	□ < /*	∞ <*l	 _		⊠ <*[₽ <u>₽₽₽</u>]*<*
	- II	⊠ □ <*]*	□ ⊠ <=	⊠ □*<	□ ⊠ *	□ Ø <]*	□ <u>∞</u> <*/		2 <	₽,	₩ *-*
[] 23 <	8 [*<	⊠⊡ <*%*	□ ~ ~*	₹[]	, , ₪	□Ø <%*	⊠ [] <*]	□ _⊠ *] <	IJ. ₽₹	[*<*

Fig. 9. The 48 spatial relationships of category disjoint.

	₩ * T	لھے *ا[□ ⊒ [[*1	/*i*	-₩]*	፼፼፼ * [] [₽ *]*
-0	≟₿	* #	 =	₩ */	2222]* *	□Ø %*	⊠ * =	822 %	₩ *%*
]]* I		L∎∓ T	₽`) 1	%* *	□ [*		₽ 2 [₩ * [*
□ 8 %	₩ %*	= *	□ /*	₩ *%	272223 [* *	₽]/	₽ * /*	=	7 22 3 / *

Fig. 10. The 40 spatial relationships of category join.

][*	□ []]%*]]*	[* [1	□ 223 /[= /	Ø /* /]* /	
×] ∦	₩ ₩ ₩	₩ %]*	[*%	/]	∎22a ∕%	₩ * *	[* /	□ 22 /*]	<u>/////////////////////////////////////</u>
₽ [[*	₩ ₩ ₩ ₩		[*	□ 1 22 /=	//*	⊠∎ /* %	፼ ₽ /* [⊠ □ /*=	[/*
] *[፼፼]* %	2222]*]	- 22 /]*	⊐ 22 ∕%*	-₩2 / [*	Ø /* /*	22	Ø P /*%*	= /*
∞ %* [₽ <mark>,</mark> %*%	22 2 2 %*]]/	% /	[/	Ø⊟ ∕* [*]/*	₩ %/*	//]* /*

Fig. 11. The 50 spatial relationships of category part_ovlp.

==	=%] =]%	□∎ □ %=	* %	[=	[%
=]	= [][* %]	— %[

Fig. 12. The 16 spatial relationships of category contain.

	=%*	Ø]*=	2111]*%*	2 2 %*=	%*%*	[* =	[*%*
=]*	=[*	[]*]*	///]*[*	%*]*	%*[*	[*]*	[* [*

Fig. 13. The 16 spatial relationships of category belong.

the Prime-Number-based Matrix (PN Matrix) representation.

3.1. Characteristics of spatial categories

Now, we will describe our observation of characteristics of those five spatial categories as follows. Suppose A and B are two objects in a picture f, and the spatial relationship between them in terms of x- and y-axis is $(Ar_{A,B}^x B, Ar_{A,B}^y B)$, where $r_{A,B}^x$ and $r_{A,B}^y$ are the spatial operators in Table 2.

- 1. Disjoin: One or both the $r_{A,B}^x$, $r_{A,B}^y$ spatial operators are in $\{<, <^*\}$.
- 2. Join: (a) None of the $r_{A,B}^x$, $r_{A,B}^y$ spatial operators is in $\{<,<^*\}$. And (b) one or both of the $r_{A,B}^x$, r_{AB}^y spatial operators are in $\{|,|^*\}$.

- 3. Contain: Both the $r_{A,B}^x$, $r_{A,B}^y$ spatial operators are in $\{=, \%,], [\}.$
- 4. Belong: Both the $r_{A,B}^x$, $r_{A,B}^x$ spatial operators are in $\{=,\%^*,]^*,[^*\}.$
- 5. Part_overlap: (a) One of the $r_{A,B}^x$, $r_{A,B}^y$ spatial operators is in $\{/,/^*\}$ and the other is in $\{\%, [,], /, =, \%^*, [^*,]^*, /^*\}$. Or (b) one of the $r_{A,B}^x$, $r_{A,B}^{y}$ spatial operators is in $\{\%,], [\}$ and the other is in $\{\%^*,]^*, [^*\}$.

To make those category rules more clear, we transform them into more formal descriptions. Suppose the spatial relationship between objects A and B is $(Ar_{A,B}^{x}B, Ar_{A,B}^{y}B)$, then the spatial category of A and B is described as follows:

- 1. Disjoin: $(r_{A,B}^{x} \in \{<,<^{*}\})$ or $(r_{A,B}^{y} \in \{<,<^{*}\})$. 2. Join: (a) $((r_{A,B}^{x} \notin \{<,<^{*}\})$ and $(r_{A,B}^{y} \notin \{<,<^{*}\})$), and (b) $((r_{A,B}^{x} \in \{|,|^{*}\})$ or $(r_{A,B}^{y} \in \{|,|^{*}\})$). 3. Contain: $(r_{A,B}^{x} \in \{=,\%,[,]\})$ and $(r_{A,B}^{y} \in \{=,\%,[,]\})$. 4. Belong: $(r_{A,B}^{x} \in \{=,\%^{*},[^{*},]^{*}\})$ and $(r_{A,B}^{y} \in \{=,\%^{*},[^{*},]^{*}\})$
- [*,]*}).
- 5. Part_Overlap: (a) $(r_{A,B}^{x}(r_{A,B}^{y}) \in \{/, /^{*}\} \text{ and } r_{A,B}^{y}(r_{A,B}^{x}) \in \{\%, [,], =, /, \%^{*}, [^{*},]^{*}, /^{*}\}), \text{ or } (b) (r_{A,B}^{x}(r_{A,B}^{y}) \in \{\%, [,], =, /, \%^{*}, [^{*},]^{*}, /^{*}\})$ $\in \{\%,], [\}$ and $r_{A,B}^{y}(r_{A,B}^{x}) \in \{\%^{*},]^{*}, [^{*}\}.)$

3.2. Assignments of spatial-operator-values (SOV) for 13 spatial operators

Based on the above observation, we can support efficient spatial reasoning by making use of the Prime-Number-based Matrix Strategy. The major step is to assign each spatial operator a unique number according to these five spatial categories. Suppose r is a spatial operator in the set $\{<, <^*, |, |^*, [, [^*,],]^*, \%, \%^*, /, /^*, =\}$ and A, B are two objects in the symbolic picture. We define sov(r) as the spatial-operator-value of r with a initial value 1, and $sv(A, B) = sov(r_{A,B}^x) \times sov(r_{A,B}^y)$ as the spatial-value of the object pair (A, B). Here comes the steps of assignments.

To classify the disjoin category, the prime number 2 is applied. That is,

$$\operatorname{sov}(<) \coloneqq \operatorname{sov}(<) \times 2$$
, $\operatorname{sov}(<^*) \coloneqq \operatorname{sov}(<^*) \times 2^2$.

Since only sov(<) and $sov(<^*)$ are the multiples of 2, one $(sv(A, B) \mod 2)$ operation can determine whether one or both of the $r_{A,B}^x$, $r_{A,B}^y$ spatial operators are in the set $\{<,<^*\}$. That is, one $(sv(A, B) \mod 2)$ operation can determine the disjoin category.

To classify the join category, the prime number 3 is applied. That is,

$$\operatorname{sov}(|) \coloneqq \operatorname{sov}(|) \times 3, \quad \operatorname{sov}(|^*) \coloneqq \operatorname{sov}(|^*) \times 3^2.$$

Since only sov(|) and $sov(|^*)$ are multiples of 3, one $(sv(A, B) \mod 3)$ operation can determine whether one or both of the $r_{A,B}^x$, $r_{A,B}^y$ spatial operators are in the set $\{|, |^*\}$. Moreover, none of the $r_{A,B}^x$, $r_{A,B}^y$ spatial operator should be in the set $\{<,<^*\}$, so one $(sv(A, B) \mod 2)$ operation is needed. That is, one $(sv(A, B) \mod 2)$ operation and one $(sv(A, B) \mod 3)$ operation can determine the *join* category.

To classify the contain category, the prime number 5 is applied. We multiply 5 to the spatial-operator-value of the spatial operators which are in the set $\{=, \%,], [\}$. That is,

$$\operatorname{sov}(r) \coloneqq \operatorname{sov}(r) \times 5, \quad \forall r \in \{=, \%,], [\}.$$

From the observation of the contain category, one $(sv(A, B) \mod 5^2)$ operation can determine whether both the $r_{A,B}^x$, $r_{A,B}^y$ spatial operators are in the set $\{=, \%,], [\}$. That is, one $(sv(A, B) \mod 5^2)$ operation can determine the contain category. But to distinguish these four symbols, three more prime numbers must be applied. Therefore,

$$\operatorname{sov}(\%) \coloneqq \operatorname{sov}(\%) \times 7, \quad \operatorname{sov}(]) \coloneqq \operatorname{sov}(]) \times 11,$$

 $\operatorname{sov}([) \coloneqq \operatorname{sov}(]) \times 13.$

To classify the belong category, the prime number 17 is applied. We multiply 17 to the spatial-operatorvalue of the spatial operators in the set $\{=, \%^*,]^*, [^*\}$. That is,

$$\operatorname{sov}(r) \coloneqq \operatorname{sov}(r) \times 17, \quad \forall r \in \{=, \%^*, |*, [*]\}.$$

From the observation of the contain category, one $(sv(A, B) \mod 17^2)$ operation can determine whether both the $r_{A,B}^x$, $r_{A,B}^y$ spatial operators are in the set $\{=, \%^*,]^*, [^*\}$. That is, one $(sv(A, B) \mod 17^2)$ operation can determine the contain category. But to distinguish these four symbols, three more prime numbers must be applied. Therefore,

$$\operatorname{sov}(\%^*) \coloneqq \operatorname{sov}(\%^*) \times 19, \quad \operatorname{sov}(]^*) \coloneqq \operatorname{sov}(]^*) \times 23,$$

 $\operatorname{sov}([^*) \coloneqq \operatorname{sov}([^*) \times 29.$

To classify the part_overlap category, let us consider the following two cases stated before. First, to determine whether one of the $r_{A,B}^x$, $r_{A,B}^y$ spatial operators is in the set $\{/,/^*\}$, and the other is in the set $\{\%, [,], /, =, \%^*, [^*,]^*, /^*\}$, two prime numbers 31 and 37 are applied. That is,

$$\operatorname{sov}(/) \coloneqq \operatorname{sov}(/) \times 31, \quad \operatorname{sov}(/^*) \coloneqq \operatorname{sov}(/^*) \times 31^2,$$

 $\operatorname{sov}(r) \coloneqq \operatorname{sov}(r) \times 37, \quad \forall r \in \{\%, [,], /, =, \%^*, [^*,]^*, /^*\}.$

Therefore, one $(sv(A, B) \mod (31 \times 37^2))$ operation can determine whether one of the $r_{A,B}^x$, $r_{A,B}^y$ spatial operators is in the set $\{/, /^*\}$, and the other is in the set $\{\%, [,], /, =, \%^*, \%^*, [^*,]^*, /^*\}$. That is, one (sv(A, B))mod (31×37^2)) operation can determine the first case of the part_overlap category.

Second, to determine whether one of the $r_{A,B}^x$, $r_{A,B}^y$ spatial operators is in the set $\{\%,], [\}$, and the other is in the set $\{\%^*,]^*, [^*\}$, two prime numbers 41 and 43 are applied. That is,

$$sov(r) \coloneqq sov(r) \times 41, \quad \forall r \in \{\%,], [\};$$

$$sov(r) \coloneqq sov(r) \times 43, \quad \forall r \in \{\%^*,]^*, [^*\}.$$

Therefore, one $(sv(A, B) \mod (41 \times 43))$ operation can determine whether one of the $r_{A,B}^x$, $r_{A,B}^y$ spatial operators is in the set $\{\%,], [\}$, and the other is in the set $\{\%^*, [^*,]^*\}$.

That is, one $(sv(A, B) \mod (41 \times 43))$ operation can determine the second case of the *part_overlap* category. Consequently, one $(sv(A, B) \mod (31 \times 37^2))$ operation and one $(sv(A, B) \mod (41 \times 43))$ operation can determine the *part_overlap* category.

According to the above descriptions, we have assigned each spatial operator a unique value which can be used to determine different spatial categories efficiently. However, in order to determine the spatial relationships between any two objects efficiently, we have to make each of the spatial-operator-values of these spatial operators indivisible by the spatial-operator-value of any other spatial operator. That is, no one spatial-operatorvalue of a spatial operator is a multiple of the spatialoperator-value of any other spatial operator. Therefore, we let $sov(<) \coloneqq sov(<) \times 47$ to distinguish spatial operators < and <*, let $sov(|) := sov(|) \times 53$ to distinguish spatial operators "|" and "|*", and let $sov(/) \coloneqq sov(/) \times 59$ to distinguish spatial operators "/" and "/*". Finally, the assignments of spatialoperator-values for these 13 spatial operators are shown in Fig. 14, and the five category rules based on the module operation are shown in Fig. 15

3.3. Data structure for pictorial symbol representation: The PN matrix

In the 9DLT matrix representation, the spatial relationships between each object pairs are obvious. Thus, spatial reasoning and pictorial query could work efficiently. However, it is conspicuous that a 9DLT matrix concluding the spatial relationships between two objects into nine types is insufficient. Conversely, the 2D C-string represents a picture more precisely since it concludes the spatial relationships into 169 types. However, spatial reasoning based on the 2D C-string representation is not so straightforward. Therefore, we propose a Prime-Number-based Matrix (PN Matrix) strategy to preserve the spatial information in the 2D C-string representation by using an extended 9DLT matrix which can support to answer the spatial relationship directly and support pictorial query and similarity retrieval easily.

Suppose a picture f contains m objects and let $V = \{v_1, v_2, \ldots, v_m\}$. Let A be the set of 13 spatial operators $\{<, <^*, |, |^*, [, [^*,],]^*, \%, \%^*, /, /^*, =\}$. An

<	: 2 × 47	$<^*$: 2^2
1	: 3 × 53	* : 3 ²
%	$: 5 \times 7 \times 37 \times 41$	%* : 17 × 19 × 37 × 43
[$: 5 \times 13 \times 37 \times 41$	[* : 17 × 29 × 37 × 43
]	$: 5 \times 11 \times 37 \times 41$]* : 17 × 23 × 37 × 43
1	: 31 × 37 × 59	/* : 31 ² × 37
=	$: 5 \times 17 \times 37$	

Fig. 14. The assignments of those 13 spatial operators.

Disjoin	$if sv(A, B) \mod 2 = 0.$
Join	if $sv(A, B) \mod 2 \neq 0$, and $sv(A, B) \mod 3 = 0$.
Contain	$\text{if } sv(A,B) \mod 5^2 = 0.$
Belong	if $sv(A, B) \mod 17^2 = 0$.
Part_overlap	if $sv(A, B) \mod (31 \times 37^2) = 0$, or $sv(A, B) \mod (41 \times 43) = 0$.

Fig. 15. Category rules.

 $m \times m$ spatial matrix S of picture f is defined as follows:

$$S = \begin{bmatrix} v_1 & v_2 & \cdots & v_{m-1} & v_m \\ 0 & r_{1,2}^y & \cdots & \cdots & r_{1,m}^y \\ r_{1,2}^x & 0 & \ddots & \vdots \\ \vdots & \ddots & 0 & \ddots & \vdots \\ \vdots & \ddots & 0 & r_{m-1,m}^y \\ r_{1,m}^x & \cdots & r_{m-1,m}^x & 0 \end{bmatrix}$$

where the lower triangular matrix stores the spatial information along the x-axis, and the upper triangular matrix stores the spatial information along the y-axis. That is, $S[v_i, v_j] = r_{j,i}^x$ if i > j; $S[v_i, v_j] = r_{i,j}^y$ if i < j; $S[v_i, v_j] = 0$ if i = j, $\forall v_i, v_j \in V$, $\forall r_{j,i}^x, r_{i,j}^y \in A$, $1 \le i, j \le m$, where $r_{j,i}^x$ is the spatial operator between objects v_i and v_j along the x-axis and $r_{i,j}^y$ is the spatial operator between objects v_i and v_j along the y-axis. Note that in this representation, we always record the relationships between two objects v_i and v_j from the view point of object v_i no matter along the x-axis or the y-axis, where i < j. That is why $S[v_i, v_j] = r_{ji}^x$ when i > j.

For the pictorial picture shown in Fig. 16, the corresponding spatial matrix S is shown as follows:

	Α	В	С	D	Ε	
AB	0		%	%* *	/*]	
$S = \begin{array}{c} C \\ D \\ E \end{array}$	< < * * %	0 <* /*	<* 0 <* <*	%* %* 0 <	<* /* [0	

According to the assignments of spatial-operator-values for those 13 spatial operators described before, we can transform the spatial matrix S of f into a PN matrix T by replacing each spatial operator $r_{j,i}^x$ $(r_{i,j}^y)$ with its unique spatial-operator-value as follows:



Fig. 16. An image and its symbolic representation.

$$S = \begin{bmatrix} AB & C & D & E \\ 03 \times 53 & 5 \times 7 \times 37 \times 41 & 17 \times 19 \times 37 \times 43 & 31^2 \times 37 \\ 2 \times 47 & 0 & 2^2 & 17 \times 19 \times 37 \times 43 & 2^2 \\ 2 \times 47 & 3 \times 53 & 0 & 17 \times 19 \times 37 \times 43 & 31^2 \times 37 \\ 2^2 & 2^2 & 2^2 & 2^2 & 0 & 5 \times 13 \times 37 \times 41 \\ 17 \times 19 \times 37 \times 43 & 31^2 \times 37 & 2^2 & 2 \times 47 & 0 \end{bmatrix}$$

3.4. Spatial reasoning based on the PN matrix representation

Spatial reasoning means the inference of a consistent set of spatial relationships among the objects in an image. Based on the PN matrix, it is easy to retrieve the spatial relationships of each pair of objects along the x-axis and y-axis straightforwardly, since this information is recorded directly in the matrix. Moreover, the category of each pair of objects can be inferred by following the category rules as shown in Fig. 15, in which only one or two module operations on the sv value (the spatial value) are needed.

3.5. Pictorial query based on the PN matrix representation

A pictorial query allows the users to query images with a specified spatial relationship. For example "display all images with a lake east of a mountain". In this section, we will describe the pictorial query processing based on the PN matrix representation according to the query types classified by Lee and Hsu.⁽¹⁴⁾ First, the following basic orthogonal directional aggregates are the main body of a PN matrix query:

- 1. x: (east, A, B) iff $r_{A,B}^x \in \{<^*, |^*, [,\%, /^*\}$ (i.e. iff sov($r_{A,B}^x) \in \{2^2, 3^2, 5 \times 13 \times 37 \times 41, 5 \times 7 \times 37 \times 41, 31^2 \times 37\}$).
- 2. y: (north, *A*, *B*) iff $r_{A,B}^{y} \in \{<^{*}, |^{*}, [, \%, /^{*}\}$ (i.e. iff sov $(r_{A,B}^{y}) \in \{2^{2}, 3^{2}, 5 \times 13 \times 37 \times 41, 5 \times 7 \times 37 \times 41, 31^{2} \times 37\}$).
- 3. x: (west, A, B) iff $r_{A,B}^x \in \{<, |, [, \%, /\}$ (i.e. iff sov $(r_{A,B}^x) \in \{2 \times 47, 3 \times 53, 5 \times 13 \times 37 \times 41, 5 \times 7 \times 37 \times 41, 31 \times 37 \times 59\}$).
- 4. $y: (\text{south}, A, B) \text{ iff } r_{A,B}^{y} \in \{<, |, [, \%, /\} \text{ (i.e. iff sov}(r_{A,B}^{y}) \in \{2 \times 47, 3 \times 53, 5 \times 13 \times 37 \times 41, 5 \times 7 \times 37 \times 41, 31 \times 37 \times 59\}).$

Therefore, the primitive direction relationship problem becomes a membership checking of a set of numbers. Then, the pictorial queries based on a PN matrix can be processed as follows:

(A) Orthogonal direction object queries. For this class of queries, we still have to follow the same 15 spatial rules as described by Lee and Hsu.⁽¹⁴⁾ For example, to determine which object is in the east-north of object X, the following rule is used: $(ne, ?, X) = \{A \mid (north, A, X) \cap (east, A, X)\}.$

(B) Category relationship object queries. This class of queries allows the users to retrieve the objects with a

specified category and object, for example, "find those objects which are disjoin with object A". Based on PN matrix representation, this class of queries can be easily answered by applying the module operation as shown in Fig. 15, where the constant B is replaced with a variable X to denote the unknown object.

(C) Auxiliary relationship object queries. This class of queries allows the users to retrieve the objects with a specified auxiliary relationship and an object, where auxiliary relations contain *same*, *surround* and *part_surround*, and two symmetric-inverse relationships *surrounded* and *part_surrounded*. We can make use of sov of spatial operators to process this class of queries.

(a) A is the same as B, if A is at the same location as B along western, eastern, southern, and northern directions. That is, A is the same as B iff both $r_{A,B}^x$ and $r_{A,B}^y$ are = [i.e. $sv(A, B) \mod (5^2 \times 17^2 \times 37^2) = 0$].

(b) A surrounds B, if A contains B, and A completely surrounds B along four orthogonal directions. That is, A surrounds B iff both $r_{A,B}^x$ and $r_{A,B}^y$ are % [i.e. sv(A, B) mod $(5^2 \times 7^2 \times 37^2 \times 41^2)=0$].

(c) A part_surrounds B, if A contains B, and A surrounds B along two or three orthogonal directions. That is, A part_surrounds B iff A contains B, A is not the same as B and A does not surround B [i.e. $sv(A, B) \mod 5^2 = 0$, $sv(A, B) \mod (5^2 \times 17^2 \times 37^2) \neq 0$, and $sv(A, B) \mod (5^2 \times 7^2 \times 37^2 \times 41^2) \neq 0$].

(D) Icon relationship object queries. This type of query allows the users to retrieve all objects with a specified icon in Fig. 4. For example, to retrieve the objects whose spatial relationships with object A along the x- and y-axes are "/" and "|" respectively, a set of objects S will be retrieved, where $S = \{B \mid \text{sov}(r_{A,B}^x) \mod (31 \times 37 \times 59) = 0 \text{ and sov}(r_{A,B}^y) \mod (3 \times 53) = 0\}$.

(E) *Icon relationship queries*. This type of query allows the users to retrieve the spatial relation icon in Fig. 4 with two specified objects, for example, "find the spatial relationship icon of objects A and B". It is clear that the work is similar to the above query class (D).

(F) Category relationship queries. This type of query can answer the spatial category according to two given objects. To find the spatial category with objects A and B, category rules described in Section 3.4 are used. For example, A contains B iff $sv(A, B) \mod 5^2 = 0$.

(G) Orthogonal direction queries. Finally, the orthogonal spatial relationship between objects A and B can be examined. The four orthogonal directional aggregates described above are used.

3.6. Similarity retrieval based on the PN matrix representation

The target of similar retrieval is to retrieve the images that are similar to the query image. In this section, we will describe the similar types and the corresponding similarity retrieval algorithm based on our PN matrix strategy. Basically, we will show how those similar types which are defined in the 2D C-string representation⁽⁴⁾ can be determined based on our PN matrix representation. Now, the definitions of the similar types⁽¹⁴⁾ are described as follows:

Definition 1. Picture f' is a type-*i* unit picture of f, if (1)f' is a picture containing the two objects A and B, represented as $x : Ar_{A,B}^{x'}B$, $y : Ar_{A,B}^{y'}B$, (2) A and B are also contained in f and (3) the relationships between A and B in f are represented as $x : Ar_{A,B}^{x}B$, and $y : Ar_{A,B}^{y}B$, then,

(type-0): Category
$$(r_{A,B}^{x}, r_{A,B}^{y}) = Category(r_{A,B}^{x'}, r_{A,B}^{y'});$$

(type-1): (type-0) and $(r_{A,B}^{x} = r_{A,B}^{x'})$ or $r_{A,B}^{y} = r_{A,B}^{y});$
(type-2): $r_{A,B}^{x} = r_{A,B}^{x'}$ and $r_{A,B}^{y} = r_{A,B}^{y};$

where $Category(r_{A,B}^x, r_{A,B}^y)$ denotes the relationship category of the spatial relationship as shown in Fig. 4.

Definition 2. Given a matrix M_1 , a matrix operator R is defined as follows:

$$M=(M_1)^R,$$

where

$$M(i,j) = M_1(i,j) * M_1(j,i), \quad 1 \le i \le m, \ 1 \le j < i.$$

Definition 3. Given two matrices M_1 and M_2 , a matrix operator -' is defined as follows:

 $M = M_1 - M_2$, where

$$M(i,j) = M_1(i,j) - M_2(i,j), \quad \forall \ 1 \le j \le i \le n.$$

Definition 4. Given a $m \times m$ PN matrix T, the corresponding category matrix C is defined as follows.



Fig. 17. One example of pictures f_1 and f_2 .

used to determine whether two pictures are of type-0, type-1, type-2 similarity, respectively, given two PN matrices T_1 and T_2 .

Algorithm (type-0)

- 1. $T'_1 = (T_1)^R$, $T'_2 = (T_2)^R$.
- 2. Following the category rules, find the category matrix C_1 and C_2 representing the two pictures f_1 and f_2 , respectively.
- 3. $C = C_1 C_2$. If C is zero in the lower triangular matrix, these two pictures are of type-0 similarity; otherwise, there is no match.

Algorithm (type-1)

- 1. Algorithm (type-0) passed.
- 2. $T = T_1 T_2$.
- 3. $T^* = (T)^R$. If T^* is zero in the lower triangular matrix, these two pictures are of type-1 similarity; otherwise, there is no match.

Algorithm (type-2)

1. $T = T_1 - T_2$. If T is zero, these two pictures are of type-2 similarity; otherwise, there is no match.

Now, we use one example to show how those algorithms work. Consider the pictures as shown in Fig. 17.

Step 1. Find the spatial matrices S_1 and S_2 and the PN matrices T_1 and T_2 representing the two pictures f_1 and f_2 ,

$$C[i,j] = \begin{cases} 1 & \text{if } (T[i,j] \times T[j,i]) \mod 2 = 0; \\ 2 & \text{if } (T[i,j] \times T[j,i]) \mod 2 \neq 0, \text{ and } (T[i,j] \times T[j,i]) \mod 3 = 0; \\ 3 & \text{if } (T[i,j] \times T[j,i]) \mod 5^2 = 0; \\ 4 & \text{if } (T[i,j] \times T[j,i]) \mod 17^2 = 0; \\ 5 & \text{if } (T[i,j] \times T[j,i]) \mod (31 \times 37^2) = 0, \text{ or } (T[i,j] \times T[j,i]) \mod (41 \times 43) = 0, 1 \le i \le m, 1 \le j < i. \end{cases}$$

That is, C[i,j] = 1, 2, 3, 4, 5 if the relationship between objects v_i and v_j is of the disjoin, join, contain, belong and part_overlap category, respectively, by following the category rules.

Based on these two new matrix operators, R and -', the following three algorithms, type-0, type-1, type-2 are

respectively.

$$S_{1} = \begin{bmatrix} A & B & C & D \\ B & & & \\ C & & \\ D & & \\ D & & \\ \end{bmatrix}, \begin{bmatrix} 0 & /^{*} & /^{*} & <^{*} \\ \% & 0 & / & <^{*} \\ < & < & 0 & <^{*} \\ / & / & \%^{*} & 0 \end{bmatrix},$$

A prime-number-based matrix strategy for efficient iconic indexing

$$T_{1} = \begin{matrix} A & B & C & D \\ 0 & 31^{2} \times 37 & 31^{2} \times 37 & 2^{2} \\ 5 \times 7 \times 37 \times 41 & 0 & 31 \times 37 \times 59 & 2^{2} \\ 2 \times 47 & 2 \times 47 & 0 & 2^{2} \\ 31 \times 37 \times 59 & 31 \times 37 \times 59 & 17 \times 19 \times 37 \times 40 & 0 \end{matrix} \right],$$

$$A = \begin{matrix} A & B & C & D \\ 0 & A & B & C & D \end{matrix}$$

$$S_{2} = \frac{A}{D} \begin{bmatrix} 0 & /^{*} & / & <^{*} \\ / & 0 & | & <^{*} \\ C & < & 0 & <^{*} \\ / & \%^{*} &]^{*} & 0 \end{bmatrix}$$

$$T_{2} = \begin{bmatrix} A & B & C & D \\ 0 & 31^{2} \times 37 & 31 \times 37 \times 59 & 2^{2} \\ 31 \times 37 \times 59 & 0 & 31 \times 53 & 2^{2} \\ 2 \times 47 & 2 \times 47 & 0 & 2^{2} \\ 31 \times 37 \times 59 & 17 \times 19 \times 37 \times 43 & 17 \times 23 \times 37 \times 43 & 0 \end{bmatrix}$$

Step 2. Calculate T'_1 and T'_2 .

$$T_{2}' = (T_{2})^{R} = \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} \begin{bmatrix} 31^{3} \times 37^{2} \times 59 \\ 2 \times 31 \times 37 \times 47 \times 59 \\ 2^{2} \times 31 \times 37 \times 59 \end{bmatrix} \begin{bmatrix} 2 \times 3 \times 47 \times 53 \\ 2^{2} \times 17 \times 19 \times 37 \end{bmatrix}$$

Step 3. Following the category rules, compute the corresponding category matrices, where 1, 2, 3, 4 and 5 mean the disjoin, join, contain, belong and part_overlap relationship, respectively.

$$A \quad B \quad C \quad D$$

$$C_{1} = \begin{pmatrix} A \\ B \\ C \\ D \\ D \\ \end{bmatrix} \begin{pmatrix} 5 \\ 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ \end{pmatrix},$$

$$A \quad B \quad C \quad D$$

$$C_{2} = \begin{pmatrix} A \\ B \\ C \\ D \\ D \\ \end{bmatrix} \begin{pmatrix} 5 \\ 1 & 1 \\ 1 & 1 & 1 \\ \end{bmatrix}.$$

Step 4. Check type-0 similarity. Since C = 0 in the lower triangular matrix, these two pictures are of type-0 similarity.

$$C = C_2 - C_2 = \begin{pmatrix} A & B & C & D \\ B & \\ C & \\ D & \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \end{pmatrix}.$$

Step 5. Check type-1 similarity. Since T = 0 in the lower triangular matrix, these two pictures are of type-1 similarity.

$$T = (T_1 - T_2)^R = \begin{bmatrix} A & B & C & D \\ B & \\ C & \\ D & \\ 0 & 0 & 0 \end{bmatrix}$$

Step 6. Check type-2 similarity. Since $T \neq 0$, these two pictures are not of type-2 similarity.

$$T = T_1 - T_2 \neq \begin{bmatrix} A & B & C & D \\ A & B & 0 & 0 & 0 \\ C & 0 & 0 & 0 \\ D & 0 & 0 & 0 \end{bmatrix}$$

3.7. A comparison

In this subsection, we make a comparison of our proposed strategy and the previous proposed representation strategies, which is as shown in Table 3.

The first item considers the data structure to represent the spatial relationships. Only the 9DLT matrix representation and the PN matrix representation record the spatial relationships in matrices. In this way, efficient matrix operations are applied, instead of complex string comparison strategies used for string representation. The second item considers the contents of representation. Only 9DLT matrix representation records the relationship in 2D space, instead of the relationship along x- and y-axis recorded in other proposed strategies. Recording spatial relationship in 2D space makes pictorial query easy; however, the 9DLT matrix strategy records only nine spatial relationships. The third item concerns whether a cutting mechanism is applied. Cutting mechanisms are applied in 2D G-string and 2D C-string representation strategies to handle overlapping objects. Since cuttings make an object into more than one components, more symbols will be used, which results in a requirement of large storage space and complex query processing strategies. Although the 2D string and 9DLT matrix representations do not need cutting mechanisms, less spatial relationships could be recorded. While in our proposed strategy, we do not need the cutting mechanisms, but we still can handle the cases of overlapping objects. The fourth item shows the number of spatial relationships in each of the proposed strategies. Only

nine spatial relationships can be recorded in the 2D string and 9DLT matrix representations. Although the 2D Gstring and 2D C-string representation strategies can record 169 spatial relationships, they need cutting mechanisms. Our proposed strategy is the only one that can record 169 spatial relationships without cutting.

The fifth item concerns whether it is easy or difficult to do spatial reasoning, where spatial reasoning means the inference of a consistent set of spatial relationships among the objects in an image. In general, if a cutting mechanism is used in the representation, then it is hard to do spatial reasoning as described in Section 2.3. The sixth item considers the way to process a pictorial query, where a pictorial query allows the users to query images with a specified spatial relationship. The 2D string representation strategy processes a pictorial query by using string matching. The 2D C-string representation strategy process pictorial queries by using complex rank rules and computing rules. The 9DLT matrix representation strategy processes pictorial queries by using matrix minus operations. Our proposed strategy process pictorial queries by using the module operation, which is more efficient as compared to the strategies used in the other representations. The last item shows the number of similar types which a representation strategy can distinguish. Basically, our proposed strategy can distinguish the same three similar types defined in the 2D C-sting representation strategy.

4. CONCLUSION

Picture indexing techniques are needed to make easy pictorial information retrieval from a pictorial database. In this paper, we have proposed an efficient iconic indexing scheme called *Prime-Number-based Matrix* (PN Matrix) for symbolic pictures, which combines the advantages of the 2D C-string and the 9DLT matrix.

Table 5. A comparison								
	2D string	2D G-string	2D C-string	9DLT matrix	Proposed			
Representation	String	String	String	Matrix	Matrix			
Contents of representation	Relationship along x-axis, along y-axis	Relationship along x-axis, along y-axis	Relationship along x-axis along y-axis	Relationship in 2D space	Relationship along <i>x</i> -axis, along <i>y</i> -axis			
With cutting?	No	Yes	Yes	No	No			
Relationship types	9	169	169	9	169			
Spatial reasoning	Easy	Hard	Hard	Easy	Easy			
Pictorial query	By using string matching	Not proposed	By using rank rules and computing rules	By using matrix minus operations	By using mod operations			
Similar types	3	Not proposed	3	Not proposed	3			

Table 3. A comparison

In the proposed scheme, we have designed each spatial operator a unique value which is a product of some prime numbers, and derived five category rules. Since those category rules are module-operation-based, they are efficient enough as compared to the previous approaches. We have also proposed a Prime-Number-based Matrix data structure to represent pictorial data, in which the relationship between two objects is recorded obviously. Consequently, spatial reasoning can be done very straightforwardly. For answering a pictorial query, some module operations or membership checking of a set of numbers have been applied. In similarity retrieval, some new matrix operations have been applied. A picture is defined to be ambiguous if there exists more than one different reconstructed picture from its representation. How to handle the problem of an ambiguous picture is one future research direction. Furthermore, how to efficiently handle large amounts of image databases is also an important future research direction.

REFERENCES

- S. K. Chang, C. W. Yan, D. C. Dimitroff and T. Arndt, An intelligent image database system, *IEEE Trans. Software Eng.* 14, 681–688 (May 1988).
- S. K. Chang, Q. Y. Shi and C. W. Yan, Iconic indexing by 2-D strings, *IEEE Trans. Pattern Analysis Mach. Intell.* PAMI-9, 413–428 (May 1987).
- G. Costagliola, F. Ferrucci, G. Tortora and M. Tucci, Correspondences—non-redundant 2D strings, *IEEE Trans. Knowledge Data Eng.* 7, 347–350 (April 1995).

- S. Y. Lee and F. J. Hsu, 2D C-string: a new spatial knowledge representation for image database systems, *Pattern Recognition* 23, 1077–1087 (October 1990).
- P. W. Huang and Y. R. Jean, Using 2D C⁺-string as spatial knowledge representation for image database systems, *Pattern Recognition* 27, 1249–1257 (September 1994).
- S. Y. Lee, M. K. Shan and W.P. Yang, Similarity retrieval of iconic image database, *Pattern Recognition* 22, 675–682 (June 1989).
- 7. S. S. Liaw, A new scheme for retrieval similar pictures in a pictorial database, *J. Comput.* 2, 19–26 (December 1990).
- E. Jungert, Extended symbolic projections as a knowledge structure for spatial reasoning, *Proc. Fourth BPRA Conf.* on *Pattern Recognition*. Springer, Cambridge (March 1988).
- E. Jungert and S. K. Chang, An algebra for symbolic image manipulation and transformation, *Visual Database Systems*, T. S. Kunii, ed., pp. 301–307. North-Holland, Amsterdam (1989).
- S. K. Chang, E. Jungert and Y. Li, Representation and retrieval of symbolic pictures using generated 2D strings, Technical Report, University of Pittsburgh, PA 15260 (1988).
- C. C. Chang, Spatial match retrieval of symbolic pictures, J. Inform. Sci. Eng. 7, 405–422 (December 1991) (1993).
- C. C. Chang and J. C. Lin, A fast spatial match accessing scheme for symbolic pictures, J. Electrical Eng. 33, 129– 137 (June 1990).
- C. C. Chang and H. C. Wu, A module-oriented signature extraction to retrieve symbolic pictures, J. Comput. 2, 45– 54 (December 1990).
- S. Y. Lee and F. J. Hsu, Spatial reasoning and similarity retrieval of images using 2D C-string knowledge representation, *Pattern Recognition* 25, 305–318 (March 1992).

About the Author — YE-IN CHANG was born in Taipei, Taiwan, in 1964. She received the B.S. degree in Computer Science and Information Engineering from National Taiwan University, Taipei, Taiwan, in 1986, and the M.S. and Ph.D. degrees in Computer and Information Science from the Ohio State University, Columbus, Ohio, in 1987 and 1991, respectively. She joined the Department of Applied Mathematics, National Sun Yat-Sen University, Kaohsiung, Taiwan, in 1991, where she is now an Associate Professor. Her research interests include database systems, distributed systems, knowledge-based systems and computer networks.

About the Author — BI-YEN YANG was born in Taipei, Taiwan, R.O.C., in 1971. She received the B.S. degree and M.S. degree in Applied Mathematics from National Sun Yat-Sen University in Taipei, Taiwan, in 1994 and 1996, respectively. Her research interests include database systems and computer networks.