COMMUNICATIONS IN INFORMATION AND SYSTEMS
Vol. 3, No. 3, pp. 203-222, March 2004

005

A STEPWISE APPROXIMATION OF INTRACTABLE SPATIAL CONSTRAINTS IN IMAGE QUERIES*

QING-LONG ZHANG† AND STEPHEN S.-T. YAU‡

Abstract. An image stored in image database systems is assumed to be associated with some content-based meta-data about that image, that is, information about objects in the image and absolute/relative spatial relationships among them. An image query for such an image database system can generally be handled in two ways: exact picture matching and approximate picture matching. We address the approximate picture matching problem of central interest in this paper, and present a stepwise approximation of intractable spatial constraints in an image query. Especially, this stepwise approximation may be pre-processed on an image query before an advanced picture matching algorithm is invoked. We then work out details of the stepwise approximation algorithm by analyzing, one by one, all possible 16 cases for results of the object matching step. Our analysis turns out that only 13 cases are valid, while the other 3 cases are identified impossible for finding an exact picture-matching between a query picture and a database picture. The worst-case running time complexity is given for each of them. In order to reduce the number of database pictures being matched by a user query, we also provide two suggestions to help enhance the effectiveness of image retrieval at the additional time cost.

1. Introduction.

Image database systems have been very active over the past 20 years. With the explosive interest for the last 10 years in multimedia systems, content-based image retrieval has attracted the attention of researchers across several disciplines, including computer vision, pattern recognition, human-computer interaction, and image databases. Applications that use image databases include office automation, computer-aided design, robotics, art museums, geographic information systems, remote sensing and management of earth resources, law enforcement and criminal investigation, large medical image databases, digital library, and defense. Existing methods on image representation and retrieval can be found in the literature (see, e.g., [1, 2, 6, 7, 8, 10, 11, 23]). A recent survey on content-based image retrieval was given by Smeulders et al. [12].

Tanimoto [14] suggested the use of picture icons as picture indexes, thus introducing the concept of iconic indexing. Subsequently, Chang et al. [4] developed the concept of iconic indexing by introducing the 2D string representation of the image. With the 2D string approach, the problem of pictorial information retrieval for 2D pictures becomes a problem of 2D string subsequence matching [4, 9]. Since then, the 2D string approach has been studied further in the literature (see, e.g., [3]).

Zhang [15, Chapter 3] defined two efficiently decidable classes of two-dimensional scenes, pseudo-symbolic pictures and extended pseudo-symbolic pictures, and introduced the extended pseudo-2D string representation for them. Zhang et al. [17] then extended his

*Received on July 4, 2003; accepted for publication on March 5, 2004.

†Member, IEEE. Department of Information Technology, University of South Florida, Lakeland Campus, 3433 Winter Lake Road, Lakeland, Florida 33803, USA.

‡Fellow, IEEE. Control and Information Laboratory, Department of Mathematics, Statistics, and Computer Science, University of Illinois at Chicago, 322 Science and Engineering Offices, 851 South Morgan Street, Chicago, Illinois 60607, USA. E-mail: yau@uic.edu

work on the extended pseudo-symbolic pictures to work for the whole images. In that paper [17], they proposed a unified iconic indexing, the generalized combined 2D string representation, for images in image databases. Each 2D image is modelled as a generalized extended pseudo-symbolic picture, which has the GEP-2D string representation. Their proposed iconic indexing (GC-2D string representation) combines both the GEP-2D string representation and the usual 2D string representation to capture absolute and relative spatial relationships in the image. The result is better representation of spatial relationships in image databases. Spatial relationships may be absolute or relative, and directional or topological. The 2D string approach developed by Chang et al. [4] is based on (relative) directional spatial relationships: *left-of, right-of, above*, and *below*. Spatial relationships proposed in our work [15, 16, 17, 19] are quite general, can be (absolute) directional, (relative) directional, or (absolute) topological.

In [18], Zhang et al. proposed the important consistency problem about content-based image indexing and retrieval, and presented a model for Content-based Image DataBase Systems (CIDBS). They described major components of image database systems, and demonstrated how these components work together to facilitate content-based image indexing and retrieval. In their most recent paper [20], Zhang and Yau have presented a very general approach to indexing and retrieval of images in image database systems.

The rest of this paper is organized as follows. In Section 2, we show how to specify a general user query, and then utilize our polynomial-time algorithm, proposed in [18], to verify the consistency among spatial constraints in the query image. In Section 3, we address the approximate picture matching problem of central interest in this paper, and present a stepwise approximation of intractable spatial constraints in an image query. We formulate 16 possible cases for results of the object matching step, and then work out details of the stepwise approximation algorithm by analyzing them one by one. Our analysis turns out that only 13 cases are valid, while the other 3 cases are identified impossible for finding an exact picture-matching between a query picture and a database picture. The worst-case running time complexity is given for each of them. In order to reduce the number of database pictures being matched by a user query, we also provide two ways to help enhance the effectiveness of image retrieval at the additional time cost. In Section 4, we discuss related work. Conclusion and future research are given in Section 5.

2. A Consistent Image Query Formulation. In this section, we first show how to specify a general user query, and then utilize our polynomial-time algorithm, proposed in [18], to verify the consistency among spatial constraints in the query image.

2.1. Specifying a User Query. A real picture is assumed to be associated with some content-based meta-data about that picture, that is, information about objects in the picture and absolute/relative spatial relationships among them. An object in a real picture corresponds to a significant element of the image. Depending on the application, the significant elements of the image can be pixels, lines, regions, etc. A spatial relationship among objects is relative if it is determined by the position of the centroid of its objects. A spatial relationship is absolute if it is determined by the absolute position of its objects in the image. The following various absolute spatial relationships are of common interest in pictorial databases: *left-of, right-of, in-front-of, behind, above, below, inside, outside,* and *overlaps*. Only the first six spatial operators are considered for relative spatial relationships, since *inside, outside,* and *overlaps* operators are not applicable. Note that the first six spatial operators are directional and the last three spatial operators are topological.

We will use x^a and x^r to indicate the absolute spatial operator x and the relative spatial operator x respectively. For example, below^a is the absolute *below* spatial operator while below^r is the relative *below* spatial operator. Note that *right-of* and *above* are dual spatial operators of *left-of* and *below* respectively, and *in-front-of* and *behind* spatial operators are only applicable for three dimensional pictures. Let

$$R = \{\mathit{left\text{-}of}^a, \mathit{left\text{-}of}^r, \mathit{below}^a, \mathit{below}^r, \mathit{inside},\ \mathit{outside},\ \mathit{overlaps}\}. \tag{1}$$

Clearly we can just use these seven spatial operators in R to specify spatial constraints among objects in a planar (i.e., two-dimensional) picture. Then an image stored in the image database is assumed to be represented by objects in the image and the complete information about absolute/relative spatial relationships of R among them.

Now a user query is of the following form:

(2) **An image query** Q: Find images containing a nonempty finite set O^Q of objects and another set (possibly null) F^Q of absolute/relative spatial relationships of R among them.

That is, an image f stored in the image database is matched by an image query Q if f contains a set O^Q of objects satisfying spatial constraints F^Q among these objects in O^Q. We call it *an exact picture-matching* between Q and f. Note that the set F^Q in the image query Q implicitly indicates the conjunction of all absolute/relative spatial relationships in F^Q using the $\wedge$ (i.e., AND) logical operator. A more general user query may then be formulated from the above user queries in (2) using both $\wedge$ (i.e., AND) and $\vee$ (i.e., OR) logical operators. Observe that $(Q_1 \vee Q_2) \wedge Q_3 = (Q_1 \wedge Q_3) \vee (Q_2 \wedge Q_3)$ for any three user queries Q_1, Q_2, and Q_3 in (2); and both $Q_1 \wedge Q_3$ and $Q_2 \wedge Q_3$ can be reduced, in view of the user's preference, to the appropriate forms of a user query in (2) respectively. Thus, a more general user query is, in fact, the disjunction of the above user queries in (2) using $\vee$ (i.e., OR) logical operators. Therefore, a more general user query can always be handled via its user subqueries in (2); that is, the query outcome of a more general user query consists of all query outcomes of its user subqueries in (2).

2.2. Consistency Checking of Spatial Constraints in a User Query. The meta-data about a real picture might contain certain incorrect information about the picture, which is introduced during the image capture stage, possibly because of limitations of existing image-processing algorithms or manual errors. In [18], we considered the following *consistency* problem: *Does the meta-data about a picture contain certain contradictory information across the entire database having all meta-data information about pictures?* The consistency problem is of common interest for content-based image database systems, and the Consistency Checking Mechanism is dedicated to verify/maintain consistency of meta-data information about pictures across the entire database. We [18] presented a polynomial-time algorithm for consistency checking of spatial relationships in a picture. However, our proposed consistency checking algorithm might fail to detect inconsistency of the description of absolute spatial relationships (involving *inside, outside,* and *overlaps*) for certain planar pictures.

As suggested in [18], the consistency problem may happen to an image query. It is possible to have contradictory content-based information description about a picture in an image query the user provides through the User Interface during the image retrieval stage. This may occur for a casual or novice user, and more likely when more objects are involved in an image query. This type of contradictory image queries needs to be detected/corrected during the image query formulation and before they are submitted for the Image Matching component. Otherwise, it will be a disaster if a contradictory image query is submitted for the image-matching in a huge image database, since it will waste time particularly for applications such as law enforcement and criminal investigation, and consume a lot of real-time system resources spending on image-matching which definitely results in no matching. We can utilize our polynomial-time algorithm, proposed in [18], to verify the consistency among spatial relationships in the query image. Note that it is not necessary to check the consistency among objects in the query image.

3. A Stepwise Approximation of Spatial Constraints. An image query for image databases can generally be handled in two ways: exact picture matching and approximate picture matching [20]. In this section we address the approximate picture matching problem of central interest in this paper.

Let an image query Q have q ($q \geq 0$) absolute/relative spatial constraints r_i, $1 \leq i \leq q$, among its involved objects. Observe that, when $q = 0$, no spatial constraints are imposed among objects in Q and the image matching process will only perform matching of objects between Q and a picture stored in the database. Let f be a stored image fetched from the database. To satisfy the spatial constraints imposed by Q on f during the image query processing, we can have the following stepwise query approximation: match each spatial relationship r_i ($1 \leq i \leq q$) imposed by the query image Q, one after one, in the stored image f; this process will stop and return “No Matching between Q and f” if at least one of these q spatial relationships is not matched in f, and it will successfully terminate otherwise.

The following Theorem 3.1 states that every database image matched exactly by a user query is always matched by it using the above stepwise query approximation. This important fact ensures that all database images, matched exactly by a user query, are always contained in the query outcome of the stepwise query approximation, though additional database images may be approximately matched by it.

THEOREM 3.1. *The above stepwise query approximation always terminates successfully on every stored image matched exactly by the user query.*

Proof. It is obvious. □

THEOREM 3.2. *The above stepwise query approximation on spatial constraints has the algorithmic time complexity of $O(q*N^2)$, where q is the number of absolute/relative spatial constraints in the user query and N is the number of involved objects in a stored image.*

Proof. Observe that, in order to match each spatial relationship r_i $(1 \leq i \leq q)$ imposed by the query image Q in the stored image f, one needs to check at most $O(N^2)$ spatial relationships in f. □

Theorem 3.2 shows that our proposed stepwise query approximation on spatial constraints is reasonably fast. Especially, this stepwise approximation is quite simple and straightforward to implement, it may thus be pre-processed on an image query before an advanced picture-matching algorithm is invoked.

Let Q be a user query. We can use our proposed efficient algorithm with the time complexity at most $O(m^3)$, in [15, Chapter 2] [17], to compute the set of all deducible spatial relationships from F^Q under a system of rules (i.e., the maximal set of F^Q), where m is the number of all involved objects in Q. In view of the following Theorem 3.3, we can always take the modified user query of Q, with the set of objects O^Q and the maximal set of F^Q, to perform the stepwise query approximation on spatial constraints. Clearly it helps exclude the database pictures, which fail the test of additional spatial constraints not in F^Q, from being matched by Q at the extra time cost as indicated in Theorem 3.2. Thus, it helps reduce the number of database pictures being matched by Q to enhance the effectiveness of image retrieval.

THEOREM 3.3. *Let Q and f be a user picture and a database picture respectively. If f is matched by the modified user query with the set of objects O^Q and the maximal set of F^Q using the above stepwise query approximation on spatial constraints, then f is matched by Q too.*

Proof. Directly from the fact that the maximal set of F^Q always contains F^Q. □

3.1. Matching of Objects. The picture-matching algorithm consists of both matching of objects and matching of spatial relationships [20]. Given a query picture Q and a picture f stored in the database, the matching step between objects in Q and objects in f will check all possible pairs of an object in Q and an object in f. Let Q have totally s significant objects O_k^Q $(1 \leq k \leq s)$ and f have totally t significant objects O_l^f $(1 \leq l \leq t)$. Then the step for matching of objects will yield that: each object O_k^Q $(1 \leq k \leq s)$ in Q matches a possibly empty subset of objects O_l^f $(1 \leq l \leq t)$ in f. An object O^Q in the query image Q matches an object O^f in a stored image f, denoted by $O^Q \longrightarrow O^f$, if the picture-matching process determines the matching between O^Q and O^f using an object matching scheme M_β in $\mathscr{M}$ [20]. If there exists an object O^Q in Q matching none of objects O^f in f, then the stored image f is clearly not matched by the given query image Q, and the picture-matching process will continue on the next available image stored in the database. Otherwise, that is, if every object O^Q in Q matches at least one object O^f in f, then the step for matching of spatial relationships will follow for further picture-matching between the query image Q and the stored image f.

An object may appear multiple times in a user query and a picture stored in the database. Multiple occurrences of a same object can be represented by a same object symbol associated with different subscripts [15, Chapter 3] [17]. Consider, for example, k (> 1) occurrences of a same object denoted by a symbol a, are represented, respectively, by a_0 ($= a$), $a_1, \ldots, a_{k-1}$. We always treat all these k object symbols a_j ($0 \le j \le k-1$) as "the same object symbol like a" (the main purpose of using subscripts is to capture different physical locations of multiple occurrences of a same object). In another word, when "two distinct (object) symbols a and b" is mentioned in the context, it does mean "a and b represent two distinct objects," and it does not mean "both a and b represent two occurrences of a same object." There are four combinations of a query picture Q and a database picture f, depending on single/multiple occurrence(s) of objects in Q and f:

1. Each (object) symbol occurs only once in both Q and f.
2. At least one (object) symbol occurs more than once in Q, while each (object) symbol occurs only once in f.
3. Each (object) symbol occurs only once in Q, while at least one (object) symbol occurs more than once in f.
4. At least one (object) symbol occurs more than once in both Q and f.

The following two cases of matching, between a symbol in Q and a symbol in f, are identified:

I. Each (object) symbol in Q matches at most one (object) symbol in f.

II. At least one (object) symbol in Q matches more than one distinct (object) symbols in f.

Furthermore, there are two possible consequences regarding matching results of two distinct (object) symbols in Q:

A. No two distinct (object) symbols in Q match a same (object) symbol in f.

B. At least two distinct (object) symbols in Q match a same (object) symbol in f.

Now, results of the matching step between objects in Q and objects in f vary on different combinations (totally 16 cases) of the above three types of considerations. As discussed in the next Section 3.2, in fact, only 13 cases are valid while the other 3 cases are identified impossible for finding an exact picture-matching between Q and f.

Observe that, if two distinct symbols in Q match a same symbol in f for case I.B., then these two distinct symbols in Q match exactly the same symbol in f.

Now, let us present examples to demonstrate all four possible cases I.A, I.B, II.A, and II.B. Consider, for example, a class of simplified *car* objects with only two properties (i.e., attributes) *make* and *type*:

car(*make*, *type*): *make* can be BMW and Ford, *type* can be passenger and sports.

Let f_1 be a database picture with three *car* objects, of which two are BMW passenger cars represented by a_0 ($= a$) and a_1, where a is *car*(BMW, passenger), and another one is a Ford passenger car represented by b_0 ($= b$), where b is *car*(Ford, passenger). Let Q_1 be a query picture having two *car* objects, with one BMW passenger car and another Ford passenger car. Then case I.A holds between Q_1 and f_1. Let Q_2 be a query picture having two *car* objects, with one BMW passenger car and another BMW car (the user doesn't specify whether the type of the second car is passenger or sports!). Then both objects in Q_2 match the same symbol a in f_1, that is, case I.B holds between Q_2 and f_1. Let Q_3 be a query picture having two *car* objects, with one BMW passenger car and another passenger car (the user doesn't specify whether the second car is BMW or Ford!). Then the second object in Q_3 matches both a and b in f_1, that is, case II.B holds between Q_3 and f_1. Let Q_4 be a query picture having two *car* objects, one of which is a passenger car (the user doesn't specify whether another car is BMW or Ford, and passenger or sports!). Then both objects in Q_4 match both a and b in f_1, that is, case II.B holds between Q_4 and f_1. Let f_2 be a database picture having three *car* objects, with one BMW passenger car represented by a_0 ($= a$), where a is *car*(BMW, passenger), one Ford passenger car represented by b_0 ($= b$), where b is *car*(Ford, passenger), and another BMW sports car represented by c_0 ($= c$), where c is *car*(BMW, sports). Let Q_5 be a query picture having two *car* objects, with one BMW car (the user doesn't specify whether it is passenger or sports!) and another Ford passenger car. Then case II.A holds between Q_5 and f_2. Let Q_6 be a query picture having two *car* objects, with one BMW car (the user doesn't specify whether it is passenger or sports!) and another passenger car (the user doesn't specify whether the second car is BMW or Ford!). Then the first BMW car in Q_6 matches both a and c in f_2, and the second passenger car in Q_6 matches both a and b in f_2; that is, case II.B holds between Q_6 and f_2.

Observe that, if two distinct symbols a and b in a user query Q match a same symbol in a database picture f for case II.B, then one of the following four holds:

- $match(a) \subset match(b)$
- $match(a) \supset match(b)$
- $match(a) = match(b)$
- $match(a) \cap match(b)$ is the proper subset of both $match(a)$ and $match(b)$

where $match(x)$ denotes the set of all symbols (including repetitive occurrences) in f matched by x in Q. The above three examples of Q_3 and f_1, Q_4 and f_1, and Q_6 and f_2 in the preceding paragraph, have demonstrated the validity of each possiblity in this observation. It is obvious that if two distinct symbols a and b in a user query Q match a same symbol in a database picture f for case I.B, then $match(a) = match(b)$.

3.2. Case Analysis of the Stepwise Approximation Algorithm. To work out details of the stepwise approximation of spatial constraints, we analyze, one by one, all possible 16 cases for results of the object matching step, formulated in Section 3.1. In fact, we will see that only 13 cases are valid, while the other 3 cases are identified impossible for finding an exact picture-matching between a query picture and a database picture.

Let Q be a user query with a set of m objects $\mathrm{O}^Q = \{O_1^Q, O_2^Q, \ldots, O_m^Q\}$ and a set of q absolute/relative spatial relationships $\mathrm{F}^Q = \{r_1, r_2, \ldots, r_q\}$. Let f be a database picture with a set of N objects $\mathrm{O}^f = \{O_1^f, O_2^f, \ldots, O_N^f\}$ and a set F^f of all absolute/relative spatial relationships among objects in O^f. Assume that all absolute/relative spatial relationships in f are maintained by the adjacency matrix representation [15, Chapter 2] [18]. So, it needs only $O(1)$ (i.e., constant) time to access/check a spatial relationship in f. Note that, it only needs $O(N^2)$ to generate the adjacency matrix representation of spatial relationships in f from F^f.

The picture-matching algorithm consists of both matching of objects and matching of spatial relationships [20]. Given a query picture Q and a database picture f, the matching step between objects in Q and objects in f will check all possible pairs of an object in Q and an object in f. In fact, if Q and f have an object symbol a^Q of k multiple occurrences and another object symbol b^f of l multiple occurrences respectively, then it only needs to perform the object matching between $a^Q(=a_0^Q)$ and $b^f(=b_0^f)$ instead of the object matching between the set of k multiple occurrences of a^Q, $\{a_0^Q, a_1^Q, \ldots, a_{k-1}^Q\}$, and the set of l multiple occurrences of b^f, $\{b_0^f, b_1^f, \ldots, b_{l-1}^f\}$. That is, to perform the step of object matching, it only needs to check $s*t$ pairs of s distinct objects in Q and t distinct objects in f. Thus, the number of object pairs needed to be checked for the step of object matching is always bounded by $m*N$, where m and N are the numbers of all involved objects in Q and in f respectively. It should be noted that $m \leq N$. Otherwise, if $m > N$, there are not enough objects in f being matched by objects in Q and an exact picture-matching between Q and f then become impossible. After the step of object matching, the step for matching of spatial relationships will follow for further picture-matching between the query image Q and the stored image f. We will perform a stepwise approximation of spatial constraints, instead of an exact matching between spatial relationships in Q and spatial relationships in f.

It should be quite simple to view the result of matching, between objects in a query picture Q and objects in a database picture f, using *an object matching matrix*. We can construct an object matching matrix M of $s \times t$, where s and t are the numbers of distinct objects in Q and in f respectively, to represent the result of object matching between Q and f in the following way. The rows and columns of the matrix M are, respectively, labelled by s distinct (object) symbols in Q and t distinct (object) symbols in f. Then M is the matrix with a 1 in row a^Q and column b^f if there is a matching between the object a^Q in Q and the object b^f in f and a 0 there otherwise. The object matching matrix representation may help facilitate the implementation of the algorithm in general, but it might not be needed, e.g., for the case I where each object symbol in Q matches at most one object symbol in f (since it will waste memory!).

Now it is quite straightforward to decide, given a query picture Q and a database picture f, which of the 16 cases is applicable to this pair of Q and f? This can be done in the following way. First, clearly it only needs the time, at most $O(m+N)$, to determine which of the four cases (1–4) is applicable to this pair of Q and f, by deciding single/multiple occurrence(s) of objects in both Q and f. In fact, we assume here that multiple occurrences of a same object x in Q are grouped together and represented by x_j in the list O^Q of all objects in Q (the subscript j of x means the repetitive occurrences of x). If it is not the case, then we have to spend the time $O(m^2)$ to maintain this type of order for the set O^Q of objects in Q by checking all possible pairs of m objects in Q (note that corresponding absolute/relative spatial relationships in F^Q might also be properly maintained in the time cost of same order $O(m^2)$, that is, they are enumerated in the list F^Q based on this type of order for the set O^Q of objects in Q and, by grouping all spatial relationships between two corresponding objects together). It should be noted that this type of order is assumed to be maintained for the set O^f of objects in the database picture f during the image indexing flow (see, e.g., [15, 17, 19] for our iconic indexing approach). Second, it only needs the time, at most $O(mN)$, to determine which of the two cases (I–II) is applicable to this pair of Q and f, by counting the number of $1's$ in each row of the object matching matrix defined in the preceding paragraph. In fact, this time complexity $O(mN)$ can be significantly reduced if other methods are employed. Third, it only needs the time, at most $O(mN)$, to determine which of the two cases (A–B) is applicable to this pair of Q and f, by counting the number of $1's$ in each column of the object matching matrix.

To illustrate the stepwise approximation of spatial constraints, we below analyze all possible 16 cases for results of the object matching step, one by one, based on the combination order of cases (I–II) first, then cases (1–4), and cases (A–B). The worst-case running time complexity is provided for each of them.

CASE I.1.A.

This is the simplest valid case. Let the result of the object matching step be

$$O_j^Q \longrightarrow O_{s_j}^f \text{ where } 1 \le j \le m. \tag{3}$$

Obviously it only needs the time $O(q)$ to verify, for each $r_j \in \mathrm{F}^Q$ $(1 \le j \le q)$, whether its corresponding absolute/relative spatial relationship under the object matching given by (3) is valid in the database picture f. If there exists one r_j $(1 \le j \le q)$ in F^Q whose corresponding spatial relationship under (3) is invalid in f, then the stored image f is clearly not matched by the given query image Q, and the picture-matching process will continue on the next available image stored in the database. Otherwise, that is, if, for each $r_j \in \mathrm{F}^Q$ $(1 \le j \le q)$, its corresponding spatial relationship under (3) is valid in f, then the step for matching of spatial relationships will yield that the stored image f is matched by the given query image Q.

Note that the stepwise approximation algorithm, for this case I.1.A, always generates an exact picture-matching between a query image and a database image, if there is a matching between them.

CASE I.1.B.

This case is impossible. Because at least two distinct (object) symbols in Q have to match a same (object) symbol with single occurrence in f, no exact picture-matching between Q and f can thus be made.

CASE I.2.A.

This case is impossible. Because multiple (> 1) occurrences of at least one (object) symbol in Q have to match a same (object) symbol with single occurrence in f, no exact picture-matching between Q and f can thus be made.

CASE I.2.B.

This case is impossible. In fact, the difficulty of this case I.2.B is at least the combined difficulty of both cases I.1.B and I.2.A. Thus, no exact picture-matching between Q and f can be made either.

CASE I.3.A.

Let the result of the object matching step be

$$O_j^Q \longrightarrow O_{s_j,0}^f, O_{s_j,1}^f, \ldots, O_{s_j,t_j-1}^f \text{ where } 1 \le j \le m. \tag{4}$$

In (4), we use the second subscript of O^f to indicate the situation of multiple occurrences. That is, the object symbol $O_{s_j}^f$ $(1 \le j \le m)$ has t_j (≥ 1) occurrence(s) in the database picture f, represented by, respectively, $O_{s_j,0}^f (= O_{s_j}^f), O_{s_j,1}^f, \ldots,$ and O_{s_j,t_j-1}^f. It should be noted that $\sum_{j=1}^m t_j \le N$.

The stepwise approximation algorithm now needs to determine that, given $1 \le k < l \le m$, whether there exist $0 \le u \le t_k - 1$ and $0 \le v \le t_l - 1$ such that all absolute/relative spatial relationships in F^Q between O_k^Q and O_l^Q can exactly match their corresponding ones in F^f between $O_{s_k,u}^f$ and $O_{s_l,v}^f$. If the answer is NO for some pair of k and l $(1 \le k < l \le m)$, that is, there exist $1 \le k < l \le m$ such that, for any pair of u $(0 \le u \le t_k - 1)$ and v $(0 \le v \le t_l - 1)$, at least one spatial relationship in F^Q between O_k^Q and O_l^Q cannot find its corresponding valid one in F^f between $O_{s_k,u}^f$ and $O_{s_l,v}^f$; then the stored image f is clearly not matched by the given query image Q, and the picture-matching process will continue on the next available image stored in the database. Otherwise, that is, if the answer is always YES for every pair of k and l $(1 \le k < l \le m)$, then the step for matching of spatial relationships will yield that the stored image f is matched by the given query image Q.

It is obvious that the number of valid absolute/relative spatial relationships between any two objects, in a picture, is always bounded by $|R|$ $(= 7)$, where R is the set of absolute/relative spatial operators defined in (1) of Section 2.1. Thus, the number of spatial relationships needed to be checked for this case is at most

$$\sum_{1 \le k < l \le m} |R| * t_k * t_l = \frac{1}{2}|R| \sum_{1 \le k < l \le m} 2t_k t_l \le \frac{1}{2}|R| (\sum_{j=1}^m t_j)^2 \le \frac{1}{2}|R| N^2.$$

Therefore, the total time for this case I.3.A is at most $O(N^2)$.

Note that, when $t_1 = t_2 = \cdots = t_m = 1$, this case I.3.A becomes essentially the same as case I.1.A, and the stepwise approximation algorithm always generates an exact picture-matching between a query image and a database image if there is a matching between them; however, when $t_j > 1$ for at least one j $(1 \le j \le m)$, the algorithm then can not guarantee that an exact picture-matching between a query image Q and a database image f has been generated when the algorithm yields YES for Q and f (i.e., an approximate matching between them).

CASE I.3.B.

Let the result of the object matching step be the union of following forms:

$$(5) \qquad O^Q_{j_1}, O^Q_{j_2}, \ldots, O^Q_{j_l} \longrightarrow O^f_{s,0}, O^f_{s,1}, \ldots, O^f_{s,t-1} \text{ where } 1 \leq j_1 < j_2 < \cdots < j_l \leq m.$$

For at least one of the forms (5), $l > 1$, and l distinct object symbols in Q, $O^Q_{j_1}, O^Q_{j_2}, \ldots,$ and $O^Q_{j_l}$, match a same object symbol Q^f_s with t (≥ 1) occurrence(s) in the database picture f, represented by, respectively, $O^f_{s,0} (= O^f_s), O^f_{s,1}, \ldots,$ and $O^f_{s,t-1}$. To have an exact picture-matching between Q and f, the following necessary condition for each form (5) must hold:

$$(6) \qquad l \leq t.$$

Observe that, the necessary condition (6) for each form (5) can be verified in the time $O(mN)$, by traversing the object matching matrix of Q and f column by column.

Now the step for matching of spatial relationships can be done in the following way. Let $1 \leq j < k \leq m$. If the two objects O^Q_j and O^Q_k are in the same form (5), that is,

$$\begin{aligned} &O^Q_j \longrightarrow O^f_{s,0}, O^f_{s,1}, \ldots, O^f_{s,t-1} \text{ and} \\ &O^Q_k \longrightarrow O^f_{s,0}, O^f_{s,1}, \ldots, O^f_{s,t-1}, \end{aligned}$$

then the stepwise approximation algorithm needs to determine that, whether there exist $0 \leq u \leq t-1$ and $0 \leq v \leq t-1$ $(u \neq v)$ such that all absolute/relative spatial relationships in F^Q between O^Q_j and O^Q_k can exactly match their corresponding ones in F^f between $O^f_{s,u}$ and $O^f_{s,v}$. Otherwise, that is, if the two objects O^Q_j and O^Q_k are not in the same form (5), then

$$\begin{aligned} &O^Q_j \longrightarrow O^f_{s_j,0}, O^f_{s_j,1}, \ldots, O^f_{s_j,t_j-1} \text{ and} \\ &O^Q_k \longrightarrow O^f_{s_k,0}, O^f_{s_k,1}, \ldots, O^f_{s_k,t_k-1}; \end{aligned}$$

this situation is similar to that in the case I.3.A, and the algorithm needs to determine that, whether there exist $0 \leq u \leq t_j - 1$ and $0 \leq v \leq t_k - 1$ such that all spatial relationships in F^Q between O^Q_j and O^Q_k can exactly match their corresponding ones in F^f between $O^f_{s_j,u}$ and $O^f_{s_k,v}$. If the answer is NO for some pair of j and k $(1 \leq j < k \leq m)$, then the stored image f is clearly not matched by the given query image Q, and the picture-matching process will continue on the next available database image. Otherwise, that is, if the answer is always YES for every pair of j and k $(1 \leq j < k \leq m)$, then the algorithm will yield that the stored image f is matched by the given query image Q.

Using the same argument as in Theorem 3.2, we conclude that the total time for this case I.3.B is always bounded by $O(qN^2)$.

CASE I.4.A.

Case I.4.A is essentially similar to Case I.3.B, except that multiple occurrences of a same object symbol in Q for Case I.4.A replace several distinct object symbols in Q for Case I.3.B in the left-handed side of (5). Thus, the stepwise approximation algorithm for this case can accordingly be formulated from that for Case I.3.B, and the time complexity is still bounded by $O(qN^2)$.

CASE I.4.B.

Case I.4.B is essentially similar to Case I.3.B, except that multiple occurrences of several distinct object symbols in Q for Case I.4.B are allowed in the left-handed side of (5) instead of several distinct object symbols (with single occurrence) in Q for Case I.3.B. Thus, the stepwise approximation algorithm for this case can accordingly be formulated from that for Case I.3.B, and the time complexity is still bounded by $O(qN^2)$.

CASE II.1.A.

Case II.1.A is essentially similar to Case I.3.A, except that several distinct object symbols in f for Case II.1.A replace multiple occurrences of a same object symbol in f for Case I.3.A in the right-handed side of (4). Thus, the algorithm for this case can accordingly be formulated from that for Case I.3.A, and the time complexity is at most $O(N^2)$.

CASE II.1.B.

Assume that, given $1 \leq j < k \leq m$,

$$\begin{aligned} &match(O_j^Q) = \{O_{j_1}^f, O_{j_2}^f, \ldots, O_{j_s}^f\} \text{ and} \\ (7) \qquad &match(O_k^Q) = \{O_{k_1}^f, O_{k_2}^f, \ldots, O_{k_t}^f\}. \end{aligned}$$

For at least one pair of j and k $(1 \leq j < k \leq m)$,

$$match(O_j^Q) \cap match(O_k^Q) \neq \emptyset.$$

To have an exact picture-matching between Q and f, the following necessary condition must hold:

(8)

If $match(O_j^Q) \cap match(O_k^Q) \neq \emptyset \ (1 \leq j < k \leq m),$ then $max(|match(O_j^Q)|, |match(O_k^Q)|) > 1.$

Note that, the necessary condition (8) can be verified in the time $O(mN)$, by traversing the object matching matrix of Q and f row by row. In fact, it can be done in the following way. First, compute $|match(O_j^Q)|$ $(1 \leq j \leq m)$, in the time $O(mN)$, by traversing the $m \times N$ object matching matrix of Q and f row by row and counting the number of $1's$ in each row. Second, in order to satisfy the necessary condition (8), it suffices to verify the following equivalent condition

(9) If $|match(O_j^Q)| = |match(O_k^Q)| = 1 \ (1 \leq j < k \leq m),$ then $match(O_j^Q) \cap match(O_k^Q) = \emptyset.$

The condition (9) can be verified in the time $O(m^2)$ by examining at most $\frac{m(m-1)}{2}$ pairs of objects O_j^Q and O_k^Q $(1 \leq j < k \leq m)$ with $|match(O_j^Q)| = |match(O_k^Q)| = 1$ (it needs only the constant time to see whether the only symbol in $match(O_j^Q)$ is the same only one in $match(O_k^Q)$). It should be noted that $m \leq N$ is always assumed.

Observe that, the necessary condition (8) is still not sufficient for Q and f to meet the size requirement of a possible exact picture-matching between them. Let f_3 be a database picture with three objects, of which two are *car* objects and another is a non-*car* object. The two *car* objects are one BMW passenger car represented by a_0 $(= a)$, where a is *car*(BMW, passenger), and another Ford passenger car represented by b_0 $(= b)$, where b is *car*(Ford, passenger). Let Q_7 be a query picture having three *car* objects, with one BMW passenger car represented by x, one passenger car represented by y and one more car represented by z (the user doesn't specify whether the second car y is BMW or Ford and the third car z is BMW or Ford and passenger or sports!). Then

$$match(x) = \{a\}, match(y) = \{a,b\}, \text{ and } match(z) = \{a,b\}.$$

That is, Case II.1.B holds between Q_7 and f_3. It is obvious that the necessary condition (8) holds, but the size requirement of a possible exact picture-matching between Q_7 and f_3 is still not met (since there is no way for $\{x,y,z\}$ in Q_7 matching $\{a,b\}$ in f_7). Nevertheless, the necessary condition (8) is useful to exclude the database pictures, which fail the basic size requirement imposed by (8), from further picture-matching.

Now the stepwise approximation algorithm needs to determine that, given $1 \le j < k \le m$, whether there exist $1 \le u \le s$ and $1 \le v \le t$ $(O^f_{j_u} \ne O^f_{k_v})$ such that all absolute/relative spatial relationships in F^Q between O^Q_j and O^Q_k can exactly match their corresponding ones in F^f between $O^f_{j_u}$ and $O^f_{k_v}$. If the answer is NO for some pair of j and k $(1 \le j < k \le m)$, then the stored image f is clearly not matched by the given query image Q, and the picture-matching process will continue on the next available database image. Otherwise, that is, if the answer is always YES for every pair of j and k $(1 \le j < k \le m)$, then the algorithm will yield that the stored image f is (approximately) matched by the given query image Q.

Using the same argument as in Theorem 3.2, we conclude that the total time for this case II.1.B is always bounded by $O(qN^2)$.

CASE II.2.A.

Case II.2.A is essentially similar to Case I.3.B, except that multiple occurrences of a same object symbol in Q for Case II.2.A replace several distinct object symbols in Q for Case I.3.B in the left-handed side of (5) while several distinct object symbols in f for Case II.2.A replace multiple occurrences of a same object symbol in f for Case I.3.B in the right-handed side of (5). Thus, the stepwise approximation algorithm for this case can accordingly be formulated from that for Case I.3.B, and the time complexity is still bounded by $O(qN^2)$.

CASE II.2.B.

Assume that, given two distinct object symbols $O^Q_{j_1}$ and $O^Q_{j_2}$ in Q,

$$\begin{array}{ll} & O^Q_{j_1,0}, O^Q_{j_1,1}, \ldots, O^Q_{j_1,k_1-1} \longrightarrow O^f_{u_1}, O^f_{u_2}, \ldots, O^f_{u_s} \text{ and} \\ (10) & O^Q_{j_2,0}, O^Q_{j_2,1}, \ldots, O^Q_{j_2,k_2-1} \longrightarrow O^f_{v_1}, O^f_{v_2}, \ldots, O^f_{v_t}. \end{array}$$

For at least one pair, such as $O^Q_{j_1}$ and $O^Q_{j_2}$, of distinct symbols in Q,

$$match(O^Q_{j_1}) \cap match(O^Q_{j_2}) \ne \emptyset.$$

To have an exact picture-matching between Q and f, the following necessary condition must hold:

$$\begin{aligned} &\text{For any two distinct symbols } O_{j_1}^{Q} \text{ and } O_{j_2}^{Q}, \text{ with } k_1 \text{ and } k_2 \text{ occurrences resp., in } Q,\\ &|match(O_{j_1}^{Q})| \geq k_1, |match(O_{j_2}^{Q})| \geq k_2, \text{ and}\\ &|match(O_{j_1}^{Q})| + |match(O_{j_2}^{Q})| - |match(O_{j_1}^{Q}) \cap match(O_{j_2}^{Q})| \geq k_1 + k_2. \end{aligned} \tag{11}$$

Note that, the necessary condition (11) can be verified in the following way. Let $O_{j_1}^{Q}, O_{j_2}^{Q}, \ldots,$ $O_{j_l}^{Q}$ be l ($\leq m$) distinct object symbols in Q. First, compute $|match(O_{j_k}^{Q})|$ $(1 \leq k \leq l)$, in the time at most $O(mN)$, by traversing the $l \times N$ object matching matrix of Q and f row by row and counting the number of $1's$ in each row. Second, $|match(O_{j_{k_1}}^{Q}) \cap match(O_{j_{k_2}}^{Q})|$ $(1 \leq k_1 < k_2 \leq l)$ can be computed, in the time $O(m^2N)$, with the help of an $l \times l$ square matrix X. X is initialized to all $0's$ in its entries. X is updated accordingly by traversing the $l \times N$ object matching matrix of Q and f column by column; that is, $X(k_1, k_2)$ is increased by 1 if and only if both $O_{j_{k_1}}^{Q}$ and $O_{j_{k_2}}^{Q}$ match a same object symbol in f. At most $\frac{l(l-1)}{2}$ entries in X can be increased by 1 after one of N columns of the object matching matrix is visited. Thus, the necessary condition (11) can be verified in the total time $O(m^2N)$.

The step for matching of spatial relationships in Case II.2.B can similarly be formulated from that in Case II.2.A. In fact, the only difference between Case II.2.B and Case II.2.A is that, given any two distinct object symbols $O_{j_1}^{Q}$ and $O_{j_2}^{Q}$ in Q, $match(O_{j_1}^{Q}) \cap match(O_{j_2}^{Q}) \neq \emptyset$ can hold (at least once) in Case II.2.B while $match(O_{j_1}^{Q}) \cap match(O_{j_2}^{Q}) \neq \emptyset$ cannot hold at all in Case II.2.A. In view of this difference, in order to perform matching of spatial relationships between $O_{j_1}^{Q}$ and $O_{j_2}^{Q}$ in Q, an extra effort is needed in Case II.2.B to avoid choosing the common object symbol in $match(O_{j_1}^{Q}) \cap match(O_{j_2}^{Q})$ as "two distinct objects in f" being matched by $O_{j_1}^{Q}$ and $O_{j_2}^{Q}$ respectively, while it is not needed in Case II.2.A. Thus, the stepwise approximation algorithm for this case can accordingly be formulated from that for Case II.2.A.

The time complexity for this case II.2.B is now bounded by $O(qN^2 + m^2N)$, where $O(qN^2)$ is taken for matching of spatial relationships (using the same argument as in Theorem 3.2) and $O(m^2N)$ is taken for verifying the necessary condition (11) on the basic size requirement.

CASE II.3.A.

Case II.3.A is essentially similar to Case I.3.A, except that multiple occurrences of several distinct object symbols in f for Case II.3.A replace multiple occurrences of a same object symbol in f for Case I.3.A in the right-handed side of (4). Thus, the stepwise approximation algorithm for this case can accordingly be formulated from that for Case I.3.A, and the time complexity is at most $O(N^2)$.

CASE II.3.B.

Case II.3.B is essentially similar to Case II.1.B, except that multiple occurrences of several distinct object symbols in f for Case II.3.B replace several distinct object symbols in f for

Case II.1.B in the right-handed side of (7). Thus, the stepwise approximation algorithm for this case can accordingly be formulated from that for Case II.1.B, and the time complexity is still bounded by $O(qN^2)$.

CASE II.4.A.

Case II.4.A is essentially similar to Case I.3.B, except that multiple occurrences of a same object symbol in Q for Case II.4.A replace several distinct object symbols in Q for Case I.3.B in the left-handed side of (5) while multiple occurrences of several distinct object symbols in f for Case II.4.A are allowed instead of multiple occurrences of only one same object symbol in f for Case I.3.B in the right-handed side of (5). Thus, the stepwise approximation algorithm for this case can accordingly be formulated from that for Case I.3.B, and the time complexity is still bounded by $O(qN^2)$.

CASE II.4.B.

Case II.4.B is essentially similar to Case II.2.B, except that multiple occurrences of several distinct object symbols in f for Case II.4.B replace several distinct object symbols in f for Case II.2.B in the right-handed side of (10). Thus, the stepwise approximation algorithm for this case can accordingly be formulated from that for Case II.2.B, and the time complexity is still bounded by $O(qN^2 + m^2N)$, where $O(qN^2)$ is taken for matching of spatial relationships (using the same argument as in Theorem 3.2) and $O(m^2N)$ is taken for verifying the necessary condition (11) on the basic size requirement.

Now, we have worked out details of the stepwise approximation of intractable spatial constraints by analyzing, one by one, all possible 16 cases for results of the object matching step, formulated in Section 3.1. Our analysis turns out that only 13 cases are valid, while the other 3 cases (i.e., I.1.B, I.2.A, and I.2.B) are identified impossible for finding an exact picture-matching between a query picture and a database picture. The worst-case running time complexity is also provided for each of them. In this Section 3.2, we have imposed the necessary condition on the basic size requirement, in the stepwise approximation algorithm, to exclude the database pictures, which fail the test of the necessary condition required by an exact picture-matching, from further picture-matching. It helps further reduce the number of database pictures being matched by a user query; and the stepwise approximation algorithm with this feature thus outperforms the algorithm without this feature introduced in the beginning of Section 3, at the additional time cost of $O(m^2N)$, where m and N are the numbers of all involved objects in a query image and in a database image respectively. However, as illustrated in Case II.1.B, the necessary condition on the basic size requirement is still not sufficient for a query image and a database image to meet the size requirement of a possible exact picture-matching between them.

In summary, we have the following Theorem 3.4.

THEOREM 3.4. *The proposed picture-matching algorithm consists of both matching of objects and matching of spatial relationships. Given a query picture Q and a database picture f, the step for matching of objects needs to check at most $m * N$ pairs of an object in Q and an object in f, where m and N are the numbers of all involved objects in Q and in f*

respectively. After the step of object matching, the step for matching of spatial relationships needs the time $O(mN)$ to decide which of the 16 cases is applicable to this pair of Q and f, and the time $O(N^2)$ to generate the adjacency matrix representation of absolute/relative spatial relationships in f for accessing/checking, in the constant time, a spatial relationship in f. Then, for performing a stepwise approximation of intractable spatial constraints, the extra algorithmic time complexity by each of 16 cases is:

(1). *$O(q)$ for Case I.1.A.*

(2). *$O(N^2)$ for each of three Cases I.3.A, II.1.A, and II.3.A.*

(3). *$O(qN^2)$ for each of seven Cases I.3.B, I.4.A, I.4.B, II.1.B, II.2.A, II.3.B and II.4.A.*

(4). *$O(qN^2 + m^2N)$ for each of two Cases II.2.B and II.4.B, where $O(m^2N)$ is taken for verifying the necessary condition on the basic size requirement.*

(5). *No extra time for each of three Cases I.1.B, I.2.A, and I.2.B, being identified impossible for finding an exact picture-matching between a query picture and a database picture.*

where q is the number of absolute/relative spatial constraints in the query picture Q.

Theorem 3.4 is a detailed version, equipped with the feature of testing the necessary condition on the basic size requirement, of Theorem 3.2. In view of both Theorems 3.2 and 3.4, the worst-case algorithmic time complexity in Theorem 3.4 is at most $O(qN^2 + m^2N)$ for both Cases II.2.B and II.4.B while it is $O(q * N^2)$ in Theorem 3.2, where the additional time cost of $O(m^2N)$ is taken for verifying the necessary condition on the basic size requirement. It should be noted that $m \leq N$ is always assumed. Observe that

$$\text{when } q \geq (\frac{m}{N})m, \text{ then } qN^2 \geq (\frac{m}{N})m * N^2 = m^2N. \tag{12}$$

Consider, e.g., $m = \sqrt{N}$ ($m = 5, N = 25$; and $m = 10, N = 100$). Then m can be considered to be much smaller than N, i.e., $m \ll N$. The condition in the observation (12) now becomes that $q \geq (\frac{m}{N})m = 1$. If m is much closer to N (i.e., $m \approx N$), then the condition in the observation (12) becomes that approximately $q \geq m$. Thus, the additional time cost of $O(m^2N)$, for both Cases II.2.B and II.4.B in Theorem 3.4, would be under $O(qN^2)$ if $m \ll N$ or approximately $q \geq m$; and it might become significant if $q \ll m \approx N$.

4. Discussion of Related Work. Costagliola et al. [5] defined three types (i.e., type-0, type-1, and type-2) of spatial relations respectively, corresponding to each of three types of picture matching by 2D strings defined in [4]. They then used these three types of spatial relations to improve the (approximate) picture-matching of type-0, type-1, and type-2, by avoiding to test some unnecessary (spatial) agreements. In their paper [5], Costagliola et al. only considered single occurrence of each symbol in the query image and single/multiple occurrence(s) of a symbol in a database image (i.e., both Cases 1 and 3 in our paper). Soffer and Samet [13] presented a method for processing pictorial queries of symbolic images from, e.g., the map domain. They considered multiple/single occurrence(s) of a symbol in a database image while requiring single occurrence of each symbol in the query image (i.e., both Cases 1 and 3 in our paper). In their concluding remarks [13], it was mentioned that Soffer and Samet were trying to work on handling multiple occurrences of a symbol in the query image. Gudivada and Raghavan [7] gave an algorithm for computing the spatial similarity between a query symbolic image and a database symbolic image, for image retrieval by spatial constraints. In fact, in their paper [7], they only considered single occurrence of each object in both query and database images (i.e., the simplest Case 1 in our paper). El-Kwae and Kabuka [6] extended the work of Gudivada and Raghavan [7], and only considered single occurrence of each object in both query and database images, too (i.e., the simplest Case 1 in our paper).

Note that, in [5, 6, 7, 13], all the authors did not consider the result of object matching between a query image and a database image (i.e., Cases I and II, and Cases A and B); in fact, they all considered only one or two case(s) (i.e., the simplest Case I.1.A, or both Cases I.1.A and I.3.A), out of totally 13 valid cases, respectively, for the questions of their concern.

In [24], Zhong and Chang used the method of Gudivada and Raghavan [7] to represent spatial relationships, and thus considered only single occurrence of each query region in a query image.

5. Conclusion and Future Research. In this paper we have first showed how to formulate a consistent general image query. Then the approximate picture matching problem of central interest in this paper is addressed, and we have presented a stepwise approximation of intractable spatial constraints in an image query. We have worked out details of the algorithm by analyzing, one by one, all possible 16 cases for results of the object matching step. Our analysis turns out that only 13 cases are valid, while the other 3 cases (i.e., I.1.B, I.2.A, and I.2.B) are identified impossible for finding an exact picture-matching between a query picture and a database picture. The worst-case running time complexity is also provided, in Theorem 3.4, for each of them.

In order to reduce the number of database pictures being matched by a user query, we can have two ways to help enhance the effectiveness of image retrieval at the additional time cost. First, given a user query Q, we can always take the modified user query of Q, with the set of objects O^Q and the maximal set of F^Q, to perform the stepwise query approximation on spatial constraints. Clearly it helps exclude the database pictures, which fail the test of additional spatial constraints not in F^Q, from being matched by Q. Second, we have imposed the necessary condition on the basic size requirement, in the stepwise approximation algorithm, to exclude the database pictures, which fail the test of the necessary condition required by an exact picture-matching, from further picture-matching. The stepwise approximation algorithm with this feature thus outperforms the algorithm without this feature.

Especially, this stepwise approximation of spatial constraints may be pre-processed on an image query before an advanced picture matching algorithm is invoked. Future research is required to further address the issue of using approximate picture matching to handle an image query in image databases. Advanced polynomial-time algorithms for the approximate picture matching need to be much developed to help improve the performance of image retrieval.

In [21], we have successfully used the method of stepwise approximation of spatial constraints to work out a similarity measure between a query image and a database image, for image retrieval. Furthermore, in [22], we have worked out an advanced picture matching algorithm to handle a general image query in our proposed content-based image database systems.

REFERENCES

[1] S. Berretti, A. Del Bimbo, and P. Pala, *Retrieval by Shape Similarity with Perceptual Distance and Effective Indexing,* IEEE Transactions on Multimedia, 2:4(2000), pp. 225–239.

[2] C. C. Chang and S. Y. Lee, *Retrieval of Similar Pictures on Pictorial Databases,* Pattern Recognition, 24:7(1991), pp. 675–680.

[3] S.-K. Chang and E. Jungert, *Symbolic Projection for Image Information Retrieval and Spatial Reasoning,* Academic Press, 1996.

[4] S.-K. Chang, Q. Y. Shi, and C. W. Yan, *Iconic Indexing by 2-D Strings*, IEEE Trans. Pattern Anal. Machine Intell., 9:3(1987), pp. 413–428.

[5] G. Costagliola, M. Tucci, and S.-K. Chang, *Representing and Retrieving Symbolic Pictures by Spatial Relations*, Visual Database Systems, II (E. Knuth and L.M. Wegner, eds.), Elsevier Science Publishers B.V. (North-Holland), pp. 49–59, 1992.

[6] E. A. El-Kwae and M. R. Kabuka, *A Robust Framework for Content-based Retrieval by Spatial Similarity in Image Databases*, ACM Transactions on Information Systems, 17:2(1999), pp. 174–198.

[7] V. N. Gudivada and V. V. Raghavan, *Design and Evaluation of Algorithms for Image Retrieval by Spatial Similarity*, ACM Transactions on Information Systems, 13:2(1995), pp. 115–144.

[8] T.-Y. Hou, P. Lui, and M. Y. Chui, *A Content-based Indexing Technique Using Relative Geometry Features*, Proceedings of SPIE – Image Storage and Retrieval Systems, the International Society for Optical Engineering, Vol. 1662, 1992.

[9] S.-Y. Lee, M.-K. Shan and W.-P. Yang, *Similarity Retrieval of Iconic Image Database*, Pattern Recognition, 22:6(1989), pp. 675–682.

[10] G. Petraglia, M. Sebillo, M. Tucci, and G. Tortora, *Virtual Images for Similarity Retrieval in Image Databases*, IEEE Transactions on Knowledge and Data Engineering, 13:6(2001), pp. 951–967.

[11] E. G. M. Petrakis, C. Faloutsos and K.-I.(D.) Lin, *ImageMap: An Image Indexing Method Based on Spatial Similarity,* IEEE Transactions on Knowledge and Data Engineering, 14:5(2002), pp. 979–987.

[12] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, *Content-Based Image Retrieval at the End of the Early Years*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 22:12(2000), pp. 1349–1380.

[13] A. Soffer and H. Samet, *Handling Multiple Instances of Symbols in Pictorial Queries by Image Similarity,* Proceedings of the First International Workshop on Image Databases and Multi-Media Search, Amsterdam, The Netherlands, pp. 51–58, August 22–23, 1996.

[14] S. L. Tanimoto, *An Iconic Symbolic Data Structuring Scheme*, Pattern Recogn. and Artificial Intell., Academic Press, New York, pp. 452–471, 1976.

[15] Q.-L. Zhang, *A Unified Framework for Iconic Indexing of Spatial Relationships in Image Databases*, Ph.D. Thesis, Department of Mathematics, Statistics, and Computer Science, Univ. of Illinois, Chicago, 1996.

[16] Q.-L. Zhang, S.-K. Chang, and S. S.-T. Yau, *A Unified Approach to Indexing Images in Image Databases*, Proceedings of the First International Workshop on Image Databases and Multi-Media Search, Amsterdam, The Netherlands, pp. 99–106, August 22–23, 1996.

[17] Q.-L. Zhang, S.-K. Chang, and S. S.-T. Yau, *A Unified Approach to Iconic Indexing, Retrieval, and Maintenance of Spatial Relationships in Image Databases*, Journal of Visual Communication and Im-

age Representation, Special Issue on Indexing, Storage, Retrieval and Browsing of Images and Video, Academic Press, 7:4(1996), pp. 307–324.

[18] Q.-L. ZHANG, S.-K. CHANG, AND S. S.-T. YAU, *The Consistency Problem on Content-based Pictorial Description in Pictorial Database Systems*, Communications in Information and Systems, International Press, 1:2(2001), pp. 225–240.

[19] Q.-L. ZHANG AND S. S.-T. YAU, *A New Iconic Indexing for 2D and 3D Scenes*, Proceedings of the 2nd Chinese World Congress on Intelligent Control and Intelligent Automation (CWCICIA 97), Xi'an, P. R. China, pp. 1667–1672, June 23–27, 1997.

[20] Q.-L. ZHANG AND S. S.-T. YAU, *A General Approach to Indexing and Retrieval of Images in Image Databases*, Communications in Information and Systems, International Press, 3:1(2003), pp. 61–73.

[21] Q.-L. ZHANG AND S. S.-T. YAU, *A Stepwise Similarity Approximation of Spatial Constraints for Image Retrieval*, In Preparation, 2003.

[22] Q.-L. ZHANG AND S. S.-T. YAU, *On the Role of Spatial Relationships in Indexing and Retrieval of Images for Large Image Databases*, In Preparation, 2003.

[23] R. ZHAO AND W. I. GROSKY, *Narrowing the Semantic Gap – Improved Text-Based Web Document Retrieval Using Visual Features*, IEEE Transactions on Multimedia, 4:2(2002), pp. 189–200.

[24] D. ZHONG AND S.-F. CHANG, *An Integrated Approach for Content-based Video Object Segmentation and Retrieval*, IEEE Transactions on Circuits and Systems for Video Technology, 9:8(1999), pp. 1259–1268.