A NEW SPATIAL RELATION MODEL FOR IMAGE INDEXING AND SIMILARITY RETRIEVAL

Ying-Hong Wang and Tai-Lung Chien

{inhon, 053672}@mail.tku.edu.tw

Department of Information Engineering, TamKang University, Tamsui, Taipei Hsien, 25137, Taiwan, R.O.C.

Abstract

In this paper, a new spatial knowledge representation model named "Two Dimension Begin-End Boundary String"(2D Bε-string) is proposed. The 2D Bε-string represents an icon by its MBR boundaries and a number of "dummy objects". The 2D Bε-string can intuitively and naturally represent the pictorial spatial information without any spatial operator. In addition, an image similarity evaluation method based on the modified "Longest Common Subsequence" (LCS) algorithm is presented. By the proposed evaluation method, not only those images which all of the icons and their spatial relationships fully accord with the query image can be sifted out, but also those images which partial icons and/or spatial relationships are similar to the query image can be applied to. It resolves the problems that the query targets and/or spatial relationships are not certain. Our representation model and similarity evaluation also simplify the retrieval progress of linear transformations, including rotation and reflection of an image.

Keyword: image retrieval, image database, spatial knowledge, spatial reasoning, similarity retrieval, 2-D Strings, LCS algorithm, 2D Bε-string

1. Introduction

In pictorial spatial application systems, there are three basic types for image indexing and retrieval: (1) by features (e.g., color, texture, shape) of the icons in images, such as the QBIC project [9] and the Virage search engine [11]; (2) by size and location of the image icons, such as R-tree [1], R*-tree [6], Quadtree [4]; (3) by relative position of the icons, such as the 2-D Strings [2] model and its variants [3,7,8,10,13,14]. The third is very suitable for those applications that do not care the actual coordinates of icon objects. For example, one may find all images which icon *A* locates at the left side and icon *B* locates at the right.

Spatial representation by 2-D Strings and its variants have gained increasing attention in a pictorial spatial database. But most of them need a series of cutting for icon objects in an image. The icons must be associated with a lot of spatial operators to accomplish the representation of related spatial information, but their representations are not intuitive. They always generate more split objects after cutting, require more space for saving, and use more complicated algorithm for image retrieval. Their similarity retrievals require massive geometric computations and focus only on those database images that own all of the icons and spatial relationships of the query image. They, however, do not mention that only part of icons and of spatial relationships are similar to the query image.

In this paper, we propose a new spatial knowledge representation model named "Two Dimension Begin-End Boundary String"(2D Bε-string). The 2D Bε-string does not need to cut any image's icons because it simply represents an icon by its MBR (Minimum Bounding Rectangle) boundaries. And by applying a number of "dummy objects", the 2D Bε-string can intuitively and naturally represent the pictorial spatial information without any spatial operators.

In addition, we also propose an image similarity evaluation method based on the modified "Longest Common Subsequence" (LCS) algorithm [5]. By our evaluation method, not only those images that all of the icons and their spatial relationships fully accord with the query image can be sifted out, but also those images which icons and/or spatial relationships are partially similar to the query image. It resolves the problems that the query targets and/or spatial relationships are not certain. It is much easier to retrieve the linear transformations of an image represented by 2D Bε-string.

The remainder of this paper is organized as follows. Section 2 reviews the approaches of 2-D Strings and its variants. In Section 3, we propose a new spatial knowledge representation model, called 2D Bε-string. A similarity retrieval algorithm and the corresponding similarity evaluation progress are introduced in Section 4. In Section 5, we present a visualized retrieval system. Finally, the conclusion and the future work are stated.

2. Related Work

Chang *et al.* [2] proposed an approach, called '2-D Strings', to represent the spatial information in a picture or image. The 2-D Strings uses the symbolic projection of a picture along the *x*-axis and *y*-axis. The 2D G-string [3], a variant of 2-D Strings, extends the spatial relationships into two sets of spatial operators and cuts all the objects along their MBR boundaries. The 2D G-string unifies the spatial relationship between two cut objects.

The 2D C-string [7, 10], another variant of 2-D Strings, leaves the leading object as a whole to minimize the number of cutting objects. It eliminates some problems associated with superfluous cutting objects generated at the 2D G-string cutting progress. It, however, is $O(n^2)$ cutting objects in the worst case.

Instead of using the cutting process, the 2D B-string [8] represents an object by two symbols in a dimension, one stands for the beginning boundary, the other for the end. The 2D B-string reduces spatial relationships to a single operator '='. It means that two objects have the same boundary projection if '=' is appeared.

0-7695-1198-8/01/$10.00 (C) 2001 IEEE

2001 IEEE International Conference on Multimedia and Expo
ISBN 0-7695-1198-8/01 $17.00 © 2001 IEEE

The basic similarity retrieval and evaluation idea for 2-D Strings [2], 2D G-string [3], 2D C-string [7] and 2D B-string [8] is the same. First, they always define three types of similarity, type-i (i = 0, 1, 2). Each is constricted by some conditions. Type-1 is stricter than type-0 and type-2 is stricter than type-1. Second, they examine all spatial relationship pairs between any two objects in the query image versus those pairs in the image of database. They build type-i subgraph if the pair satisfies type-i constraints. After examining, they find the maximum complete subgraph for each type-i graph. The number of objects in the maximum complete subgraph is the similarity of the query image and the images of database.

The space and time complexity to examine all spatial relationship pairs requires $O(n^2)$, where n is the number of object in an image. Finding maximum complete subgraph is an NP-complete problem [15]. It is a time consuming task. It is not suitable for a large number of icon objects in an image.

3. The Spatial Representation Model Using 2D Bε-string

There are many approaches proposed to represent an icon in an image, such as MBR [3,7,8,10], MBE (Minimum Bounding Ellipse) and MBC (Minimum Bounding Circle). The approach used in the 2D Bε-string is MBR. Conceptually, it is similar to the 2D B-string. However, the 2D Bε-string adopts a quite different idea to address the spatial relationship between two boundary symbols. The 2D B-string uses a **spatial operator** (=) to describe the projection of two boundaries that is **IDENTICAL**. In the 2D Bε-string, we use a **dummy object** to describe the projection of two boundaries that is **DISTINCT**!

We define a 'Dummy Object' in the following way:

A 'Dummy Object' is not a real object in the original image. It can be specified as any size of space and be memorized as symbol 'ε'.

The 2D Bε-string with n icons thus can be defined as

$$(u, v) = (d_0x_1d_1x_2d_2\ldots d_{2n-1}x_{2n}d_{2n},\ d_0y_1d_1y_2d_2\ldots d_{2n-1}y_{2n}d_{2n}).$$

Where d_i is a dummy object ε or a null string, i = 0, 1, … $2n$, and x_i and y_i are real icon objects that are either the beginning or the end projected boundaries on the x-axis and the y-axis, respectively. Set d_0 to ε if there is space between the beginning boundary of the leftmost (bottommost) object and the left (bottom) edge of an image. Similarly, set d_{2n} to ε if there is an interval between the end boundary of the rightmost (topmost) object and the right (top) edge of an image. For the rest, set d_i to ε if the boundary projections of x_i and x_{i+1} (y_i and y_{i+1}) are different. The 2D Bε-string showed in Figure 1, for example, is written as (u, v) = ($\varepsilon A_b\varepsilon B_b\varepsilon A_e C_b\varepsilon C_e\varepsilon B_e\varepsilon$, $\varepsilon B_b\varepsilon A_b\varepsilon B_e C_b\varepsilon C_e\varepsilon\ A_e\varepsilon$).

Observably, the 2D Bε-string has the following advantages: ***First***, the object location in original images and symbolic pictures was mapped directly. There is no operator required for representing the spatial relationship between objects. It is intuitively. ***Second***, it does not need to cut the objects of image, which simplifies the construction of image database. And the space complexity for an image with n objects in the worst and best cases is $O(n)$. It requires $4n$+1 and $2n$+1 symbols, respectively. ***Third***, it simplifies the similarity retrieval because there is no combination of the result of spatial reasoning required.

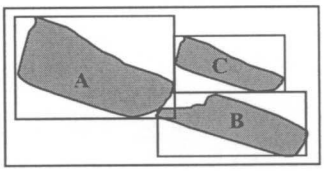

Figure 1: An image with three objects

4. Image Similarity Retrieval and Evaluation

4.1. The similarity retrieval algorithm

In the similarity assessment, we take care the number of spatial relationships formed by every two objects and appeared in the query image and database image at the same time. We propose an algorithm, as shown in Table 1, to find the LCS length from the 2D Bε-strings of a query image and database image. Then measure the similarity by evaluating this LCS string with respect to the original 2D Bε-strings.

Table 1: Algorithm to calculate the LCS length.

```
2D-Be-LCS-Length (Q, D)
1.  m←length(Q)
2.  n←length(D)
3.  // Q is a 2D Bε-string of a query image, Q = {q_i | i =
      1, 2, …, m}.
4.  // D is a 2D Bε-string of a database image, D = {d_j | j
      = 1, 2, …, n}.
5.  // W is the LCS-length inferring table, W = {w_i,j | i =
      0, 1, 2, …, m, j = 0, 1, 2, …, n}.
6.  // Initialize the first column of W by zeros.
7.  for i←1 to m do
8.      w_i,0←0
9.  // Initialize the first row of W by zeros.
10. for j←0 to n do
11.     w_0,j←0
12. // Infer each cell until all cells was evaluated.
13. for i = 1 to m do
14.     for j = 1 to n do
15. // Set current cell value.
16.         if |w_i-1,j| ≥ |w_i,j-1| then
17.             w_i,j←w_i-1,j
18.         else
19.             w_i,j←w_i,j-1
20. // Check the symbol q_i, d_j and the last symbol of LCS
      path from left-up diagonal
21.         if (q_i = d_j) and ((q_i ≠ ε) or (w_i-1,j-1 ≥ 0)) then
22. // If all are hold
23.             if (|w_i-1,j-1| +1) > |w_i,j| then
24.                 w_i,j←|w_i-1,j-1| +1
25.             if q_i = ε then
26.                 w_i,j←-w_i,j
27. return W
```

It is not necessary to examine all the spatial relationships for every two boundaries. Because **the LCS string implies that, in query image and database image, all the spatial relationships of every two boundaries in LCS string are the same.** So, the similarity can be evaluated in a reasonable time.

0-7695-1198-8/01/$10.00 (C) 2001 IEEE

2001 IEEE International Conference on Multimedia and Expo
ISBN 0-7695-1198-8/01 $17.00 © 2001 IEEE

This algorithm is modified from the LCS algorithm discussed by Cormen et al. [5]. There are two factors to revise the original LCS algorithm. First, we avoid picking dummy objects continuously because only one dummy object sufficiently represents the relative spatial relationship between two boundaries. The *if*-statement in line 21 makes this decision. Second, we omit the LCS path recording matrix by evaluating the left and up paths first and the left-up diagonal path next, as shown in lines 16-19, 23-24. The LCS path, however, can still be inferred from the matrix recording the LCS length.

The algorithm takes two 2D Bε-strings Q and D as inputs and returns the LCS-length inferring table W. In a query image with m objects, the maximum length of the 2D Bε-string in each dimension is $4m+1$. The maximum length of the 2D Bε-string of a database image with n objects is $4n+1$. The LCS-length inferring table W needs $(1+(4m+1))(1+(4n+1))$ storage units; therefore, the space complexity is $O(mn)$.

In the initialization of the first row and the first column, as shown in lines 7-8 and 10-11 in Table 1, each string symbol must be set once. The outer loop, in line 13, examines each row of W $4m+1$ times. The inner loop, in lines 14-26, examines each cell of W $(4m+1)*(4n+1)$ times. Thus, the time complexity is $O((4m+1)+(4n+2)+(4m+1)*(4n+1))$, same as $O(mn)$.

4.2. The similarity evaluation

After obtaining the LCS length and the LCS string of a query image and a database image, we need to evaluate their similarity. Conceptually, the longer the LCS string is, the more two images look similar. However, two database images with the same LCS string length do not necessarily lead to the same similarity. Therefore, we propose an assessment formula to identify the similarity of these phenomena. First, we define the following notations for the 2D Bε-string:

- N: **number of objects in a query image;**
- Q_x: **string length along x-axis in a query image;**
- Q_y: **string length along y-axis in a query image;**
- L_x: **length of LCS string with dummy objects along x-axis;**
- L_y: **length of LCS string with dummy objects along y-axis;**
- M_x: **length of L_x string without dummy object;**
- M_y: **length of L_y string without dummy object;**
- D_x: **length of boundary symbols with spatial relationships in database image based on M_x;**
- D_y: **length of boundary symbols with spatial relationships in database image based on M_y;**

Then the similarity along x-axis and y-axis are defined by

S_x: similarity along x-axis, $0 \le S_x \le 1$,

$$S_x = \begin{cases} 1-(Q_x + D_x - 2L_x)/(4N+1) & \text{if } M_x > 0, \\ 0 & \text{if } M_x = 0; \end{cases}$$

S_y: similarity along y-axis, $0 \le S_y \le 1$,

$$S_y = \begin{cases} 1-(Q_y + D_y - 2L_y)/(4N+1) & \text{if } M_y > 0, \\ 0 & \text{if } M_y = 0; \end{cases}$$

We can specify different weights, W_x and W_y, to distinguish the importance in the x-axis and y-axis. The similarity of a query image and a database image can be summary by

$$S = W_x S_x + W_y S_y, \text{ where } 0 \le S \le 1 \text{ and } W_x + W_y = 1.$$

For example, the query image in Figure 1 has three objects, so N=3. The 2D Bε-string is presented by ($\varepsilon A_b \varepsilon B_b \varepsilon A_e C_b \varepsilon C_e \varepsilon B_e \varepsilon$, $\varepsilon B_b \varepsilon A_b \varepsilon B_e C_b \varepsilon C_e \varepsilon A_e \varepsilon$) and the string length on the x-axis and y-axis, (Q_x, Q_y), is (12, 12). The database image in Figure 2 has a 2D Bε-string ($\varepsilon E_b \varepsilon A_b \varepsilon B_b \varepsilon A_e \varepsilon C_b \varepsilon F_b \varepsilon E_e \varepsilon C_e \varepsilon B_e \varepsilon F_e \varepsilon$, $\varepsilon E_b \varepsilon B_b \varepsilon E_e \varepsilon A_b \varepsilon B_e C_b \varepsilon F_b \varepsilon C_e \varepsilon A_e \varepsilon F_e \varepsilon$). The LCS strings, inferred by algorithm listed in Table 1, are $\varepsilon A_b \varepsilon B_b \varepsilon A_e C_b \varepsilon C_e \varepsilon B_e \varepsilon$ and $\varepsilon B_b \varepsilon A_b \varepsilon B_e C_b \varepsilon C_e \varepsilon A_e \varepsilon$; thus, ($L_x$, L_y) is (12, 12). The value of (M_x, M_y) can be calculated by subtracting the number of dummy objects in the LCS string from (L_x, L_y), thus, (M_x, M_y) is (6, 6). The boundary symbols with spatial relationships in a database image based on the strings of M_x and M_y are $\varepsilon A_b \varepsilon B_b \varepsilon A_e \varepsilon C_b \varepsilon C_e \varepsilon B_e \varepsilon$ and $\varepsilon B_b \varepsilon A_b \varepsilon B_e C_b \varepsilon C_e \varepsilon A_e \varepsilon$; thus, ($D_x$, D_y) is (13,12). Clearly the similarity on the x-axis and y-axis can be calculated by

$$S_x = 1-(Q_x + D_x - 2L_x)/(4N+1) = 1-(12+13-2*12)/(4*3+1) = 0.9231$$

$$S_y = 1-(Q_y + D_y - 2L_y)/(4N+1) = 1-(12+12-2*12)/(4*3+1) = 1.0$$

The weights along each axis (W_x, W_y) are given as (0.5, 0.5). Then, for the database image, the similarity is S=0.5*0.9231+0.5*1.0=0.9616.

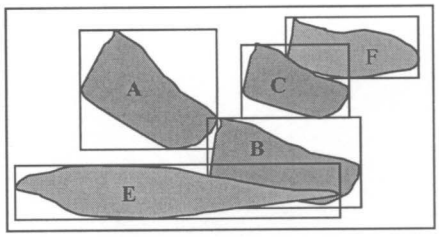

Figure 2: Database image.

4.3. The rotation and reflection of an image

If we consider the rotation (90, 180, 270 degrees clockwise) or reflection (on the x-axis or y-axis) of an image, we must perform similarity retrieval and evaluation eight times. In 2-D String, 2D G-string, and 2D C-string, we must do a proper string transformation for each dimension every time. It includes that the string may need to be reversed, and a sophisticated formula to transform spatial operators is required [12]. Even thought the 2D B-string, it still needs to recalculate their rank values.

If an image is represented by a 2D Bε-string, then the similarity retrieval of rotation and reflection for an image becomes very easy. It needs only to reverse the string if required. Then applies the similarity retrieval and evaluation progress proposed in previous sections for each transformation. Because the dummy object is not a spatial operator, its meaning is not varied while the image being rotated or reflected.

5. The Implementation

We have implemented a visual image retrieval system using the approaches of 2D Bε-string spatial representation model and the modified LCS similarity evaluation algorithm. After the retrieval system is started, the user can load an image databases from storage and browse them one by one thru the scrolling bar. The user may form the query image by putting a number of

0-7695-1198-8/01/$10.00 (C) 2001 IEEE

2001 IEEE International Conference on Multimedia and Expo
ISBN 0-7695-1198-8/01 $17.00 © 2001 IEEE

available icons in the work area, dragging them to an appropriate location, scaling them to an apposite size (Figure 3), and hit 'Search' button while completing the layout. The system will transform the query image to 2D Bε-string, and compare with the database images. Then the most similar image and its similarity information are displayed (Figure 4). One can switch to the 2D Bε-string (Figure 5) of that image by clicking on '2D Bε-string' tab. The user may browse those images next a lower degree of similarity thru the scrolling bar too.

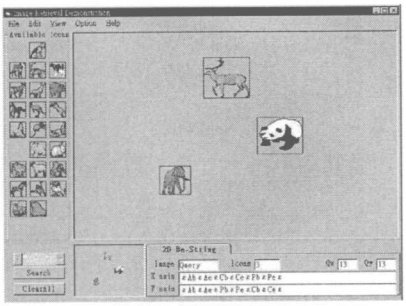

Figure 3: Arrange icons in the query image.

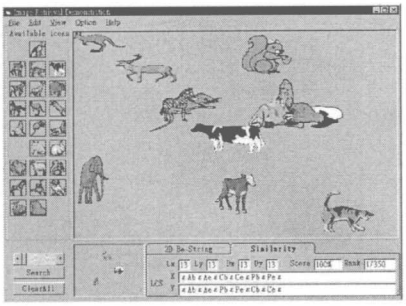

Figure 4: Show the most similar image with similarity.

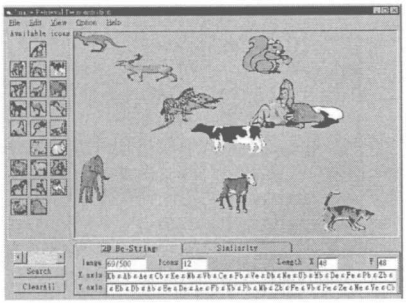

Figure 5: Show the most similar image with 2D Bε-string.

6. Conclusions and Future Work

In comparison with other 2D string methodologies, the 2D Bε-string simplifies the representation of spatial relationships and improves the efficiency of similarity retrieval. Even so, further research is still required.

At first, when an object area in an image differs very much from its MBR area, the similarity is not obvious no matter what to use, 2D Bε-string or other 2D strings. Due to the 2D Bε-string and other 2D strings obscure this information while abstracting. The distance between objects is obscured too. Therefore, how to clarify the size and distance information in the 2D Bε-string representation model is worthy studying.

In addition, we can evaluate the similarity more accurate. The original LCS algorithm does not calculate the number of LCS paths, neither do we here. For a query image from different images of database, even with the same length and content of LCS strings, conceptually, the more number of LCS paths the more similar is. Thus, we suggest further study to take account into the number of LCS paths in our similarity retrieval and evaluation approaches.

It is easy to expand the 2D Bε-string representation model and similarity assessment to retrieve objects in three-dimension case by adding the third dimension's string directly. We would like to integrate the temporal information into the video frame indexing in the near future.

References

[1] A. Guttman, "R-tree: A Dynamic Index Structure for Spatial Searching", *Proc. ACM SIGMOD Int'l Conf. On the Management of Data*, 1984

[2] S. K. Chang, Q. Y. Shi and C. W. Yan, "Iconic indexing by 2-D Strings", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. PAMI-9, No. 3, May 1987, pp.413-428.

[3] S. K. Chang, E. Jungert and Y. Li, "Representation and Retrieval of Symbolic Pictures Using Generalized 2D String", Technical Report, University of Pittsburgh, 1988.

[4] H. Samet, "Applications of Spatial Data Structures, Computer Graphics, Image Processing, and GIS." Addison-Wesley Publishing Company, 1989.

[5] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, "Introduction to Algorithms", MIT Press, 1990, pp. 314-319.

[6] N. Beckmann, H. Kriegel, R. Schneider and B. Seeger, "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles", *Proc. ACM SIGMOD Int'l Conf. On the Management of Data,* 1990

[7] S. Y. Lee and F. J. Hsu, "2D C-string: a New Spatial Knowledge Representation for Image Database Systems", *Pattern Recognition,* Vol. 23, 1990, pp. 1077-1087.

[8] S. Y. Lee, M. C. Yang and J. W. Chen, "2D B-string: a Spatial Knowledge Representation for Image Database Systems", *Proc. ICSC'92 Second Int. Computer Sci. Conf.,* 1992, pp. 609-615.

[9] W. Nibalck, R. Barber, W. Wquitz, M. Flickner, E. Glasman, D. P. P. Yanker, C. Faloutsos and G. Taubin, "The QBIC Project: Querying Images by Content Using Color, Texture and Shape", *SPIE*, 1908, 1993.

[10] P. W. Huang and Y. R. Jean, "Using 2D C^+-string as Spatial Knowledge Representation for Image Database Systems", *Pattern Recognition,* Vol. 27, No. 9, 1994, pp. 1249-1257.

[11] J. R. Bach, C. F. A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. Jain and C. Shu, "The Virage Image Search Engine: An Open Framework for Image Management", *SPIE*, 2670, 1996.

[12] B. C. Chien, "The Reasoning of Rotation and Reflection in Spatial Database", *Systems, Man, and Cybernetics*, 1998., 1998 IEEE International Conference, Vol. 2, 1998, pp. 1582–1586.

[13] Xiaobo Li and Xiaoqing Qu "Matching Spatial Relations Using DB-tree for Image Retrieval", *Pattern Recognition*, 1998., Proceedings. Fourteenth International Conference, Vol. 2, 1998, pp. 1230-1234.

[14] F. J. Hsu, S. Y. Lee and B. S. Lin, "2D C-Tree Spatial Representation for Iconic Image", *Journal of Visual Languages & Computing*, Vol. 10, No. 2, Apr 1999, pp. 147-164

[15] Michael Sipser, "Introduction to the Theory of Computation", PWS Publishing Company, 1997, pp. 245-253.

0-7695-1198-8/01/$10.00 (C) 2001 IEEE

2001 IEEE International Conference on Multimedia and Expo
ISBN 0-7695-1198-8/01 $17.00 © 2001 IEEE