

A NEW SIMILARITY METRIC BASED ON 2D VECTOR FOR ICONIC IMAGE RETRIEVAL

Peir-Cheng Chieng, and Been-Chian Chien

Department of Information Engineering,
I Shou University, Kaohsiung, Taiwan, R.O.C.
Email: cbc@csa500.isu.edu.tw

ABSTRACT

Recently, the requirements of multimedia information are increasing rapidly. The efficient similarity images retrieval from a large image data bank being the purpose of this article. The spatial relationship is an important feature in content-based retrieval. In the studying of spatial relationship, previous related works all focus on the directional relationship, such as 2D-String, 2D C-String and 2D-PIR, etc. In this paper, we show that the distance is an important factor of spatial relationship. We propose a new representation to embed the distant relationship among objects in an image and a new similarity retrieval method. The R_{norm} measure be used to evaluate the quality of our new method, and we find that the new similarity retrieval method provides efficient performance and supports flexibility for users' request.

1. INTRODUCTION

The amount of multimedia information is increasing rapidly of late years. Usually, the types of multimedia information consist of text, sound, image, animation and video. It is an important and difficult problem to find the data, which we wanted, from the vast of multimedia data bank. In traditional textual database systems, users always search text data by keywords. However, multimedia information such as an image has no keyword in it. Brief descriptions about the content of an image are given instead. Users may lose the relevant information since their descriptions or interpretations about the target image are different from the data creator's. Hence, the choice of the interpretation about an image's feature is very important.

There are many features included in an image. Those features can be concluded into two kinds: the visual feature and the relationship feature. The visual feature includes color, texture and shape of objects. There are many related works for visual feature. Chua [9] developed a "color-pair" technique for fuzzy object-level image retrieval. Mehre et al. Lu [15] proposed a quad-tree representation of color called "multilevel color histogram." The similarity between two images is computed based on their color histogram intersection. Jagadish [12] proposed shape similarity retrieval based on a two-dimensional rectilinear shape representation. Two shapes are similar if the error area is small when one shape is placed on the top of the other.

The other kind of feature is the relationship among the objects in images. In general, the relationships between two objects consist of spatiality and topology. For spatial relationships, Chang [4] proposed a symbolic representation for an iconic image called "2D-String". There are three spatial operators in the 2D-String. It is quite simple and clear. However, the three spatial operators are not sufficient to give a complete description for a picture with arbitrary complicated. Some extensions of 2D-String were proposed, such as 2D G-String and 2D C-String. An another representation contains spatial and topological relationships called 2D-Projection Interval Representation (2D-PIR) was proposed by Nabil [17]. The 2D-PIR developed a new representation and an approach of similarity measurement. The new approach of measurement allows us to retrieval images having better precision than the 2D-String.

The spatial relationships included in a picture usually consist of the directional and the distant relationships. The directional relationship describes the relative position between two distinct objects. Rather, the distant relationship stands for the length between two distinct objects. Most of the previous works about spatial relationships focus on the directional factor, though the distance is also an important factor in real applications. For directions, previous researches conclude that there are 169 directional relationships in 2D spatial plane totally. However, these directional relationships can not represent the spatial relationships of objects in 2D plane fully. For instance, the distance between objects is not recorded though it is an important component of spatiality.

In this paper, we consider not only the directional relationship but also the distant relationship. We propose a new representation for spatial relationships call 2D-Vector. A relative distance is used to describe the directional and distant relationships between two objects in an image. This paper is organized as follows. In Section 2, we review the related researches about image iconic indexing and similarity retrieval. In Section 3, the new notations and representation are introduced. The new measuring function and similarity retrieval algorithms are shown in Section 4. Then, we will show some experimental results and the comparisons with other approaches in Section 5. Finally, summaries are made and the future works are outlined in Section 6.

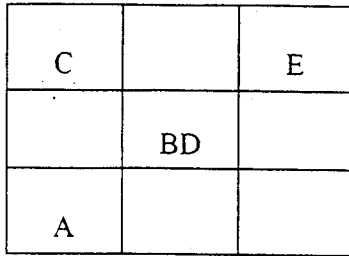


Figure 1: A symbolic picture P.

2. RELATED WORK

One of the most important problems of image database systems is image retrieval. Since there is no keyword in an image, a content-based retrieval plays a principal role for the image retrieval problem. Tanimoto [18] suggested using of picture icons as picture indexes and introducing the concept of iconic indexing. Subsequently, Chang[4] developed the 2D-String representation to be iconic indexing for iconic images. The problem of pictorial retrieval becomes a problem of 2D-String subsequence matching. This approach provides a natural way to construct iconic indices for pictures and supports spatial reasoning.

For an image, we must recognize the objects in the original image by preprocess of pattern recognition first. Assume that an object is enclosed by the minimum bounding rectangle (MBR). The centroid of a minimum bounding rectangle will be the reference point of the corresponding object. The objects are segmented by their orthogonal projection relationship respect to x - and y -axes. According to the position order of x - and y -axes, 2D-String use two symbolic strings to represent the relationships respectively.

The two strings used by 2D-String to represents the spatial relationships are u string and v string. The spatial relation operators in 2D-String include " $<$ ", " $=$ " and " $:$ ". The operator " $<$ " denotes the left-right spatial relationship in string u , or below-above spatial relationship in string v . The operator " $=$ " denotes the " $:$ " at the same spatial location as " $:$ " relationship and the operator " $:$ " stands for " $:$ in the same set as" relationship. For example, the 2D-String of a symbolic picture P as shown in Figure 1 is represented as follows: $(A=C<B:D<E, A<B:D<C=E)$. 2D-String provides a simple way to perform sub-picture matching. We can find similar pictures by matching the sub-strings on the corresponding 2D-Strings.

The spatial operators " $<$ " and " $=$ " of 2D-String are not sufficient enough to describe the spatial knowledge for pictures completely. For example, 2D-String can not represent that two objects are overlap respect to x -axis. To overcome the problems in 2D-String, Chang [5] introduced the generalized 2D-String (2D G-String) in 1989.

Chang extended the concept of symbolic projection and proposed a generalized 2D-String (2D G-String) with a cutting mechanism to describe the objects in an image. The set of generalized relational operators is {" $<$ ", " $=$ ", " $:$ "}. The edge to edge with relation operator " $|$ " is used to

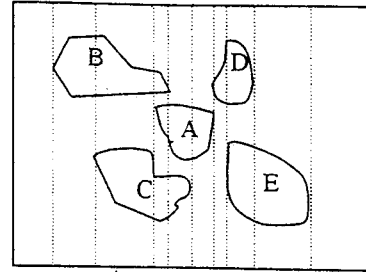


Figure 2: The partition of 2D G-String in picture P1.

solve the problem of overlapping objects. The cutting lines are performed at all the extreme points of each object in the image. Then, some objects are partitioned into many smaller subparts with the cutting lines. For example, the 2D G-String representation of picture P1 in Figure 2 is shown as follows:

2D G-String: u -string: $B|C=B|C=A=B|C=A|A|D|E=D|E$.
 2D G-String: v -string: $C=E|C=A=E|A=E|A|D|B=D$.

The 2D G-String supports more spatial relationships than 2D-String, but there still exists some problems in the reasoning rules. The segmented subparts of an object are dependent on the bounding lines of other objects. For the cases with overlapping, the storage space and computation time are costly for spatial reasoning. In order to overcome the drawbacks of 2D G-String, 2D C-String was proposed.

Lee [13] proposed a more efficient and economic cutting mechanism called 2D C-String in 1992. The 2D C-String uses 13 types of spatial location to represent all relationships between arbitrary two objects along x - (and y -) coordinate axis. There exist 169 types of spatial relationships between two minimum boundary rectangles in 2D space.

The 2D C-String employs 6 operators (" $<$ ", " $|$ ", " $:$ ", " $%$ ", " $[$ ", " $=$ ") to denote the relationships between two objects along the x - (and y -) coordinate axis. The 2D C-String representation of Figure 2 are shown as follows:

2D C-String: u -string: $B|C|A|A|C|D|E|E$.
 2D C-String: v -string: $C=E|A|A|E|B=D$.

The 2D C-String defined another three types of similarity degree to retrieve similar images.

Another more precise similar retrieval method, 2D-PIR, was proposed by Nabil [17] in 1996. It adapts three existing representation formalisms (Allen's temporal interval [1], 2D-Strings, and topological relationship [10]), and combines them in a novel way to produce the unified representation. The basic idea is that each spatial object is projected along the x and y axes forming a x -interval and y -interval for the object. Using the Allen's 13 interval relationships and together with the topological relationships, then 2D-PIR relationships are established. The 2D-PIR offers more information about spatial relationships among objects in a picture. The 2D-PIR is defined as a triple (δ, χ, ψ) , where δ is a topological relationship from the set $\{dt, to, ct, in, ov, co, eq, cb\}$ [10]. χ and ψ are interval relationships $\{., m, o, d, s, f, ., mi, oi, di, si, fi\}$. χ represents the interval relationship along the x -axis, and ψ represents the interval relationship along the y -axis. A detailed de-

scription of 2D-PIR can be found in [17].

The degree of picture similarity is dependent on the distance of two 2D-PIR relationships. Assume that there are two pictures P1 and P2 have the same objects in the pictures. We construct two 2D-PIR graphs G1 and G2 corresponding to P1 and P2 respectively. For the relationships $(\delta_1, \chi_1, \psi_1)$ in graph G1 and $(\delta_2, \chi_2, \psi_2)$ in graph G2.

The notion of "distance" between δ_1 and δ_2 is derived from the topological neighborhood graph. The distance between two relationships is the shortest paths in the neighborhood graph. The distance between two interval relationships χ_1 and χ_2 or ψ_1 and ψ_2 is defined in the same way as topological relationships. The distance between two 2D-PIR relationships, r^{G1} and r^{G2} , is defined using the following Euclidean distance formula:

$$D(r^{G1}, r^{G2}) = \sqrt{D(\delta_1, \delta_2)^2 + D(\chi_1, \chi_2)^2 + D(\psi_1, \psi_2)^2}$$

This is used to measure the difference between two corresponding relationships.

The distance between two graphs is the sum of all the distance between corresponding 2D-PIR relationships. The distance between two graphs G1 and G2 is defined by the following formula:

$$D(G1, G2) = \sum_{i=1}^n D(r_i^{G1}, r_i^{G2})$$

The 2D-PIR offers another similarity measurement method. The advantage is that 2D-PIR is more efficient than 2D String and 2D C-String. The method of 2D-PIR have good perception about similarity retrieval problem, but it need record all relationships of arbitrary two objects.

3. 2D-VECTOR

Previous related works all denote the directional relationship among objects. However, the distant relationship among objects will incur different ordering similar retrieval. We propose a new representation to embed the directional and distant relationship. This section introduces the symbols that we used in the new representation and describes the properties of the notations. We refer to the new notations as *relative distance*.

We assume that an image P consists of n objects identified by O_1, O_2, \dots, O_n . After preprocessing by image processing and pattern recognition techniques, the n objects in the original image are recognized. Each object is enclosed by a minimum bounding rectangle (MBR) with boundaries parallel to horizontal (x -) and vertical (y -) axis. Let px_i and qx_i be the two coordinates for left and right boundary of O_i 's MBR projected along x -axis, $px_i < qx_i$. py_i and qy_i are the two coordinates for up and down boundary of O_i 's MBR projected along y -axis and $py_i < qy_i$, where $1 \leq i \leq n$. The coordinates of objects' MBR in an image can use a popular measure to be a basic computing unit: for example, a pixel. The relative distance notations for a picture P containing n objects will be defined by the following definitions.

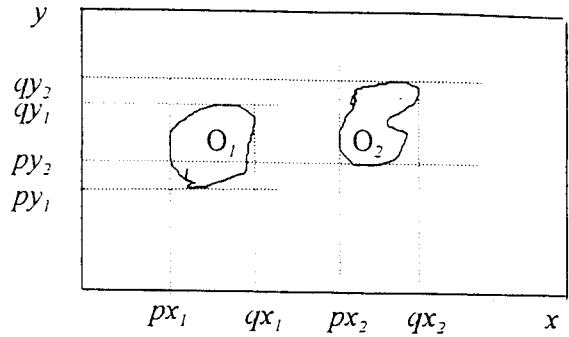


Figure 3: An example, a picture P.

Definition 1: The x - and y -axis *projection interval* of O_i are I_x and I_y , defined as

$$I_x = qx_i - px_i, \text{ and } I_y = qy_i - py_i.$$

Definition 2: Let O_i and O_j are arbitrary two objects in a picture P. We define

$$Dx_{ij} = px_j - px_i, \text{ and } Dy_{ij} = py_j - py_i.$$

Dx_{ij} and Dy_{ij} are called the x - and y -axis *distance* between O_i and O_j , referring to O_i , respectively.

Definition 3: X_{ij} is the x -axis *relative distance* between O_i and O_j , referring to O_i , $i < j$ and $X_{ij} = (Dx_{ij}/I_x, I_x/I_x)$.

Y_{ij} is the y -axis *relative distance* between O_i and O_j , referring to O_i , $i < j$ and $Y_{ij} = (Dy_{ij}/I_y, I_y/I_y)$.

Above 3 definitions have some important properties. We describe the properties in the following lemmas and prove in [8].

Lemma 1: The x - and y -axis distance are reversible; that is,

$$Dx_{ij} = -Dx_{ji} \text{ and } Dy_{ij} = -Dy_{ji}.$$

Lemma 2: If X_{ij} and Y_{ij} are defined as Definition 3, and representing $X_{ij} = (\alpha_{ij}, \chi_{ij})$, $Y_{ij} = (\beta_{ij}, \psi_{ij})$. Let $X_{ji} = (\alpha_{ji}, \chi_{ji})$ and $Y_{ji} = (\beta_{ji}, \psi_{ji})$, we have

$$\alpha_{ji} = -\alpha_{ij}, \chi_{ji} = I \chi_{ij}, \beta_{ji} = -\beta_{ij}, \psi_{ji} = I \psi_{ij}.$$

Lemma 1 shows that Dx_{ij} are the reverse direction of Dx_{ji} . Hence, the direction characteristics are embedded in the notation implicitly. The reversible property in Lemma 2 indicates that relationships between object O_i and O_j , referring to O_i , can be reasoned from the relationships referring to O_j .

Lemma 3: For n objects, let $1 \leq i, j, k \leq n$, if $X_{ij} = (\alpha_{ij}, \chi_{ij})$, $Y_{ij} = (\beta_{ij}, \psi_{ij})$ and $X_{ik} = (\alpha_{ik}, \chi_{ik})$, $Y_{ik} = (\beta_{ik}, \psi_{ik})$, then

$$X_{jk} = ((\alpha_{ik} - \alpha_{ij}), \chi_{ij}, \chi_{ik}, \chi_{ij}), Y_{jk} = ((\beta_{ik} - \beta_{ij}), \psi_{ij}, \psi_{ik}, \psi_{ij}).$$

Lemma 3 gives an important transitive property that we can derive new relationships from two indirect relationships belonging to different objects. By this property, we will reduce the quantity of stored information.

Definition 4: The 2D Vector for an image P containing n objects be represented as $\langle O_i, rx_i, ry_i \rangle$, where $rx_i = I_x, ry_i = I_y$, $rx_i = X_{ij}$, $ry_i = Y_{ij}$, $2 \leq j \leq n$ and $1 \leq i \leq n$.

An image representation should be able to describe all relationships among the objects in the image. That is, we can find the relationships between two arbitrary objects in the representation without losing information. There are $n(n-1)$

relationships totally in an image containing n objects. The information of an image we stored in the representation is

$$\langle O_1, rx_1, ry_1 \rangle, \langle O_2, rx_2, ry_2 \rangle, \langle O_3, rx_3, ry_3 \rangle, \dots, \langle O_n, rx_n, ry_n \rangle.$$

The detailed replacements with the symbols of definitions are

$$\langle O_1, lx_1, ly_1 \rangle, \langle O_2, lx_2, ly_2 \rangle, \langle O_3, lx_3, ly_3 \rangle, \dots, \langle O_n, lx_n, ly_n \rangle.$$

Thus, we can derive any relationships among the n objects by Lemma 2 and Lemma 3 based on object O_i in two computing steps at most. We refer to the representation in Definition 4 as **2D-Vector**. The first object in 2D-Vector is called **base object** or **referential object**.

Example 1: The 2D-Vector representation for the picture in Figure 2 is

$$\langle B, 6, 3 \rangle, \langle C, (0.33, 0.83), (-2.1, 1.27) \rangle, \langle A, (0.87, 0.5), (-1.03, 0.97) \rangle, \langle D, (1.37, 0.3), (-0.07, 1.1) \rangle, \langle E, (1.47, 0.7), (-2.1, 1.4) \rangle.$$

The object B is the base object. We can infer any relationships in the image by the base object. The inferential algorithm can be obtained easily from lemma 3.

4. SIMILARITY MEASUREMENT

Since the 2D-Vector preserved not only directional relationships but also distant relationships, previous measurements for pictorial similarity are not suitable for 2D-Vector. We develop new measuring methods to evaluate the similarity of pictures. In the distance aspect, the direct distance between two objects is considered first. Let O_i and O_j are two arbitrary objects in picture P1 and P2. We can compute the difference between the two objects in P1 and P2 by Euclidean distance directly. The Euclidean distance formula is defined as following:

$$D((X_{ij}, Y_{ij})^{P1}, (X_{ij}, Y_{ij})^{P2}) = \sqrt{[\alpha_{ij}^{P1} - \alpha_{ij}^{P2}]^2 + [\beta_{ij}^{P1} - \beta_{ij}^{P2}]^2}.$$

If the two pictures have the same n objects, the total difference between two pictures is defined as

$$D_{P1, P2} = \sum_{i=1}^{n-1} \sum_{j=i+1}^n D((X_{ij}, Y_{ij})^{P1}, (X_{ij}, Y_{ij})^{P2}).$$

For a user, an image retrieval system sorts the relevant images for his requirement. The users' requirements usually depend on their applications. If applications put emphasis on the directional relationship, the measurement should pay more attention on the directional measures. On the other hand, the distance should be placed. We proposed an adaptive measuring function to evaluate the difference. The adaptive function allows users to tune the weights of direction and distance. We consider three cases for measuring the similarity in the projection of x -axis. First, the case 1 that object O_i is moved from the left boundary of the picture to the left boundary of the object O_j , as Figure 4(a). In case 1 the measurement was considered that the variance become large while moving the same distance. Second, the case 2 that object O_i is moved from the right boundary of the picture to the right boundary of the object O_j , as Figure 4(b). In case 2 the measurement was considered that the variance become large while moving the

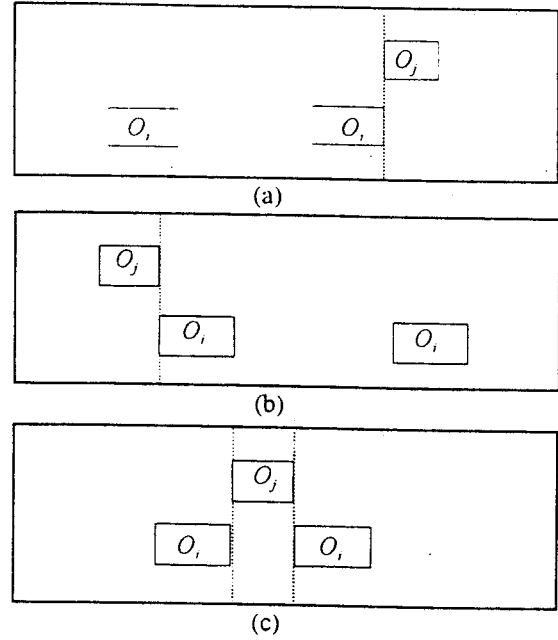


Figure 4: (a) case1: O_i moving at the left side of O_j ,
 (b) case2: O_i moving at the right side of O_j ,
 (c) case3: O_i moving from the left side of O_j to right side of O_j .

same distance. Third, the case 3 that object O_i is moving from the left boundary of object O_j to the right boundary of object O_j , as Figure 4(c). In case 3, we consider the variance is uniform. We assign a distinct measurement function to each case as shown in Figure 4(a), (b) and (c).

The relationships between two objects O_i and O_j , (X_{ij}, Y_{ij}) has a measurement function f to measure the relationships' difference with another $(X_{ij}, Y_{ij})'$, where $f \in [0, 1]$ and $\delta \in [0, 1]$. δ is a parameter to control the weight between case 1, case 2 and case 3. We can use δ to control the importance of distant relationships. The Adaptive distance formula is defined as following:

$$\text{Let } \omega(x, y) = \frac{x+y}{y+1}.$$

$$f(X_{ij}) = f(\alpha_{ij}, \chi_{ij}) = \begin{cases} \frac{\delta}{2(1 - \omega(\alpha_{ij}, \chi_{ij}))} & \text{if } \alpha_{ij} + \chi_{ij} < 0; \\ 1 - \frac{\delta}{2\omega(\alpha_{ij}, \chi_{ij})} & \text{if } \alpha_{ij} > 1; \\ (1 - \delta) \times \omega(\alpha_{ij}, \chi_{ij}) + \frac{\delta}{2} & \text{otherwise.} \end{cases}$$

The measurement function of y -axis from down to up is the same with x -axis from left to right. If two object O_i and O_j are in the different picture P1 and picture P2 then the relative relationships are $(X_{ij}, Y_{ij})^{P1}$, $(X_{ij}, Y_{ij})^{P2}$. We use the following formula to measure the difference between picture P1 and P2 on objects O_i and O_j .

$$D((X_{ij}, Y_{ij})^{P1}, (X_{ij}, Y_{ij})^{P2}) = \sqrt{[f(X_{ij})^{P1} - f(X_{ij})^{P2}]^2 + [f(Y_{ij})^{P1} - f(Y_{ij})^{P2}]^2}$$

If there are n objects in P1 and the same n objects in P2.

then the total difference between P1 and P2 is defined as follows:

$$D_{P1, P2} = \sum_{i=1}^{n-1} \sum_{j=i+1}^n D((X_{ij}, Y_{ij})^{P1}, (X_{ij}, Y_{ij})^{P2})$$

5. EXPERIMENTS AND COMPARISONS

The relevant images retrieved from databases for some queries have a ranking order for a user usually. There is no standard to judge the similarity since the results of ranking is always dependent on users' requirements. We introduce the R_{norm} measure to evaluate the quality of algorithms for similarity retrieval in image databases. The R_{norm} was proposed by P. Bollmann in 1985 [2]. The definition of R_{norm} is as following:

Definition 5 [11]: Let I be a finite set of images with a user-defined preference relation that is complete and transitive (weak order). Let Δ^{user} be the ranking order of I induced by the user preference relation. Also, let Δ^{system} be some ranking order of I induced by the similarity values computed by an image retrieval system. Then R_{norm} is defined as

$$R_{norm}(\Delta^{system}, \Delta^{user}) = \frac{1}{2} \left(1 + \frac{S^- - S^+}{S_{max}^-} \right)$$

where S^- is the number of image pairs where a better image is ranked ahead of a worse one by Δ^{system} ; S^+ is the number of pairs where a worse image is ranked ahead of a better one by Δ^{system} ; and S_{max}^- is the maximum possible number of S^- from Δ^{user} . It should be noted that the calculation of S^- , S^+ , and S_{max}^- is based on the ranking of image pairs in Δ^{system} relative to the ranking of corresponding image pairs in Δ^{user} . The range of R_{norm} value is a real number between 0 and 1.0, and the value of 1.0 indicates that the system-provided ranking order of the images is identical to the ranking order of the images provided by the user.

Given a set of 12 images as shown in Figure 5, each of them contains two objects with different spatial locations (both direction and distance). These images are numbered by P0, P1, ..., and P11. Now, we take images P0, P2, P4, P6, P8, and P10 as the query images in turn, and measure the similarity using the discussed algorithms. The algorithms of similarity retrieval we used include 2D C-String, 2D-PIR, the Euclidean distance of 2D-Vector, and the adaptive functions of 2D-Vector. The results of similarity measure are ranked by their difference values for various query images, as shown in the tables from Table 2 to Table 6.

The 2D C-String just sorts the relevant images into three types, so it has less efficient. Table 2 is the ranking results of 2D-PIR, the values under the image number in the table are the difference values between query image and the ranked images. Using the same presentation, Table 3 is the ranking results of 2D-Vector's measuring with the Euclidean distance. The measures of 2D-Vector using the adaptive functions are shown in the tables from Table 4 to Table 6. The distinctions of the results in Table 4, Table 5

and Table 6 are the weights of δ . The value of δ is 0.10 in Table 4, $\delta = 0.66$ in Table 5, and finally the value of δ is 0.90 in Table 6.

From the ranking order tables, we can find that the ranking orders of 2D-PIR are different from the Euclidean distance measuring extremely. The measures of adaptive functions also behave some different ranking results depending on the values of δ . As our prediction, the ranking order is close to the ranking results of 2D-PIR when δ is small, e.g. $\delta=0.10$. On the contrary, the ranking order is close to the ranking results of Euclidean distance when δ is large. For example, the ranking orders in Table 3 are similar to the results in Table 6 when δ is 0.90 in our experiments.

For evaluating the effectiveness of 2D-Vector more detailed, we apply the R_{norm} measure to the above results. As we know, the objective of similarity measuring is to satisfy users' requirements. However, it is difficult to show a standard ranking order for users' requirements. The ranking order for someone is hard to fit the need of everyone. According to the definition of R_{norm} measure, we develop an objective measurement to evaluate the effectiveness of 2D-Vector. Assume that there are some users whose requirements are identical to the ranking orders of the methods including the 2D-PIR and the 2D-Vector with different values of δ , respectively. These users' ranking orders are used to be the user's preference in R_{norm} measure, referred as the experts. Then, the experts are used to measure the R_{norm} values of various systems. In our experiments, we choose the same methods to be the experts and the measured systems at the same time.

We find that the R_{norm} value is close to 1 when the value δ in adaptive function is small under the expert of 2D-PIR. Contrariwise, the R_{norm} value is close to 0 when the value δ in adaptive function become large under the expert of the 2D-Vector with Euclidean Distance. Such results may explain that the 2D-PIR is measured by the features of direction and the 2D-Vector with Euclidean distance emphasizes on the distance. For users, their adjudgement is compounded of the features of direction and distance usually. For example, we can give the ranking orders for the above query images visually. The ranking results in Table 1 are made by ourselves intuitively. We refer to the results as the ranking order of a user. Comparing the R_{norm} values of the user with 2D-PIR and 2D-Vectors, it is easy to find that the best similarity occurs when δ is 0.66 from Figure 6. Thus, the value $\delta=0.66$ can be chosen to as the weight value of the adaptive function for the user. Of course, other users may use different δ value of adaptive function to satisfy their requirements.

6. CONCLUSIONS

Content-based retrieval has become a primary technique for image querying. The spatial relationship is one of the main content of features images. Previous researches as 2D-String, 2D C-String and 2D-PIR, all discussed the directional relationships and ignored distance. In this paper, we propose the 2D-Vector representation to present the

spatial relationships including the information of direction and distance. We show that the 2D-Vector can provide spatial reasoning and similarity retrieval easily. The time complexity of spatial reasoning and similarity retrieval algorithms are $O(1)$ and $O(n^2)$, respectively. The space complexity is $O(n)$ for storing an image with n objects. It is more efficient than 2D C-string and 2D-PIR. By the experiments, we also find that the accuracy is better than the previous approaches. Moreover, the new algorithm provides a flexible measure for retrieving similar images.

By our new notation, there are many related researches could be outlined. One of the extensions is in video data. Video data are organized by a sequence of related images called frames. Applying the 2D-Vector to describe the motions of a sequence of objects in temporal is under investigating. Another important future work is the spatial indexing for 2D-Vector, it is worth further studying.

7. REFERENCES

[1] J. F. Allen, "Maintaining Knowledge About Temporal Intervals." *Comm.ACM*, vol. 26, no. 11, pp.832-843, 1983.

[2] P. Bollmann, F. Jochum, Reiner, V. Weissmann and H. Zuse. The Live-Project —Retrieval experiments based on evaluation viewpoints. *In Proceedings of the 8th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, pp. 213-214, 1985.

[3] S. K. Chang, C. W. Yan, D. C. Dimitroff, and T. Arndt, "An intelligent image database system." *IEEE Trans. Software Eng.*, vol. SE-14, pp.681-688, May 1988.

[4] S. K. Chang, Q.Y. Shi, and C.W. Yan, "Iconic Indexing by 2D-String." *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.9, no. 3, pp. 413-428, May 1987.

[5] S. K. Chang, E. Jungert and Y. Li, Representation and retrieval of symbolic Pictures using generalized 2D-String. *SPIE Proc. Visual Communications and Image Processing, Philadelphia*, 5-10 November, pp. 1360-1372, 1989.

[6] C. Y. Chen and C. C. Chang, "An Object-Oriented Similarity Retrieval Algorithm for Iconic Image Databases." *Pattern Recognition Letters*, vol. 14, no. 6, pp. 465-470, June 1993.

[7] B. C. Chien and P. C. Chieng, "A New Approach of Similarity Measurement for Iconic Image Retrieval". *International Conference on Computational Intelligence and Multimedia Applications*, 1998.

[8] B. C. Chien and P. C. Chieng, "Spatial Reasoning and Retrieval of Pictures Using Relative Distances." *in Proceedings of 1997 International Symposium on Multimedia Information Processing*, 11-13, Dec. Taipei, Taiwan, pp. 221-225.

[9] T. S. Chua, S. K. Lim, and H. K. Pung, "Content-Based Retrieval of Segmented Images." *Proc. Second Ann. ACM Multimedia Conf.*, 1994.

[10] M. J. Egenhofer, "Point-Set Topological Spatial Relations." *Int'l J. Geographical Information Systems*, vol. 5, no. 2, pp.161-174, 1991.

[11] V. N. Gudivanda, and V. V. Raghavan, "Design and Evaluation of Algorithms for Image Retrieval by Spatial Similarity." *ACM Transactions on Information Systems*, Vol. 13, No. 2, April 1995, pp. 115-144.

[12] H.V. Jagadish, "Aretrieval Technique for Similar Shape." *in Proc. ACM SIGMOD 1991 Int'l Conf. Management of Data*, pp. 208-217, May 29-31, 1991.

[13] S. Y. Lee and F. J. Hsu, "Spatial Reasoning and Similarity Retrieval of Images Using 2D C-String Knowledge Representation." *Pattern Recognition*, vol. 25, no. 3, pp.305-318, 1992.

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

[15] H. Lu, B. C. Ooi, and K. L. Tan, "Efficient Image Retrieval by Color Content." Dept. of Information Systems and Computer Science, National Univ. of Singapore, 1994.

[16] R. Mehrota and J. E. Gary, "Feature-Based Retrieval of Similar Shaps." *Proc. Ninth Int'l Conf. Data Engineering*, pp. 108-115, 1993.

[17] M. Nabil and A. H. H. Ngu and J. Shepherd, "Picture Similarity Retrieval Using the 2D Projection Interval Representation." *IEEE Transactions on Knowledge And Data Engineering*, Vol. 8, 4, pp.533-539, August 1996.

[18] S. L. Tanimoto, "An iconic symbolic data structuring scheme". *in Pattern Recognition and Artificial Intelligence*, pp. 452-471, Academic Press, New York, 1976.

Query Picture	Rank ordering of expert respect to user viewpoint (Images shown within the same cell of a row have same relevance)											
P0	P0	P2	P1	P4	P3	P5, P6	P7	P10	P8	P9	P11	
P2	P2	P0	P4	P3	P1	P5	P6	P7	P10	P8	P9	P11
P4	P4	P2	P0, P5, P6			P3	P7	P10	P1	P8	P9	P11
P6	P6	P5	P7	P4, P10		P2	P0, P8		P3, P9, P11			P1
P8	P8	P10	P9	P11	P5	P6	P7	P4	P2	P3	P0	P1
P10	P10	P5	P6	P9	P8	P11	P4	P7	P2	P3	P0	P1

Table 1: The rank ordering of user's viewpoint.

Query Picture	2D-PIR ranking order and the result difference value (Images shown within the same cell of a row have same relevance)						
	P0	P0, P1 0.000	P2 2.000	P3, P4 4.000	P6, P7 6.000	P5 6.324	P9, P10, P11 7.211
P2	P2 0.000	P0, P1, P3, P4 2.000		P6, P7 4.000	P5 4.472	P9, P10, P11 5.656	P8 7.211
P4	P3, P4 0.000	P2 2.000	P0, P1, P6, P7 4.000		P5 4.472	P9, P10, P11 5.656	P8 7.211
P6	P6, P7 0.000	P5 2.000	P2, P3, P4, P9, P10, P11 4.000			P0, P1, P8 6.000	
P8	P8 0.000	P9 2.000	P5, P10, P11 4.000		P6, P7 6.000	P2, P3, P4 7.211	P0, P1 8.485
P10	P10, P11 0.000	P5, P9 2.000	P6, P7, P8 4.000		P2, P3, P4 5.656	P0, P1 7.211	

Table 2: The ranking order of 2D-PIR.

Query Picture	Ranking order of 2D-Vector with Euclidean distance. (Images shown within the same cell of a row have same relevance)											
	P0	P0 0.000	P2 0.484	P4 1.043	P5 2.024	P6 2.100	P10 2.441	P7 2.561	P3 2.640	P8 2.646	P1 3.917	P9 4.461
P2	P2 0.000	P0 0.484	P4 0.824	P6 1.753	P5 1.754	P7 2.077	P3 2.236	P10 2.280	P8 2.652	P1 4.081	P9 4.252	P11 4.338
P4	P4 0.000	P2 0.824	P5 0.980	P0 1.043	P6 1.094	P10 1.456	P8 1.888	P7 2.067	P3 2.680	P9 3.443	P11 3.579	P1 4.894
P6	P6 0.000	P5 0.398	P10 1.063	P4 1.094	P7 1.643	P2 1.753	P8 1.923	P0 2.100	P11 2.607	P9 2.637	P3 2.747	P1 5.777
P8	P8 0.000	P10 0.921	P5 1.530	P4 1.888	P6 1.923	P9 2.307	P0 2.646	P2 2.652	P11 2.839	P7 3.545	P3 4.479	P1 6.529
P10	P10 0.000	P5 0.708	P8 0.921	P6 1.063	P4 1.456	P9 2.030	P2 2.280	P11 2.320	P0 2.441	P7 2.706	P3 3.763	P1 6.343

Table 3: The ranking order of 2D-Vector with Euclidean distance.

Query Picture	Ranking order of the 2D-Vector with adaptive function ($\delta=0.10$). (Images shown within the same cell of a row have same relevance)											
	P0	P0 0.000	P1 0.037	P2 0.118	P3 0.444	P4 0.478	P7 0.911	P6 0.913	P5 0.915	P10 1.030	P11 1.040	P9 1.260
P2	P2 0.000	P0 0.118	P1 0.147	P3 0.327	P4 0.360	P7 0.794	P6 0.796	P5 0.798	P10 0.928	P11 0.935	P9 1.180	P8 1.210
P4	P4 0.000	P3 0.045	P2 0.360	P7 0.434	P6 0.436	P5 0.440	P0 0.478	P1 0.507	P11 0.636	P10 0.637	P9 0.951	P8 1.000
P6	P6 0.000	P7 0.020	P5 0.085	P4 0.436	P11 0.446	P10 0.468	P3 0.470	P2 0.796	P9 0.839	P8 0.908	P0 0.913	P1 0.943
P8	P8 0.000	P9 0.075	P10 0.441	P11 0.464	P5 0.823	P6 0.908	P7 0.929	P4 1.000	P3 1.040	P2 1.210	P0 1.280	P1 1.320
P10	P10 0.000	P11 0.031	P9 0.372	P5 0.383	P8 0.441	P6 0.468	P7 0.488	P4 0.637	P3 0.682	P2 0.928	P0 1.030	P1 1.060

Table 4: The ranking order of the 2D-Vector with adaptive function ($\delta=0.10$).

Query Picture	Ranking order of the 2D-Vector with adaptive function ($\delta=0.66$). (Images shown within the same cell of a row have same relevance)											
	P0	P0 0.000	P2 0.085	P4 0.215	P1 0.245	P3 0.255	P5 0.418	P6 0.428	P7 0.433	P10 0.493	P8 0.551	P11 0.62
P2	P2 0.000	P0 0.085	P4 0.156	P3 0.172	P1 0.29	P7 0.349	P5 0.354	P6 0.355	P10 0.448	P8 0.533	P11 0.565	P9 0.636
P4	P4 0.000	P2 0.156	P3 0.197	P5 0.203	P0 0.215	P6 0.219	P7 0.261	P10 0.294	P8 0.400	P11 0.410	P1 0.445	P9 0.484
P6	P6 0.000	P5 0.075	P7 0.136	P10 0.218	P4 0.219	P11 0.259	P3 0.283	P2 0.355	P9 0.384	P8 0.398	P0 0.428	P1 0.639
P8	P8 0.000	P10 0.188	P9 0.195	P11 0.276	P5 0.324	P6 0.398	P4 0.400	P7 0.531	P2 0.533	P0 0.551	P3 0.588	P1 0.794
P10	P10 0.000	P11 0.142	P5 0.147	P8 0.188	P9 0.191	P6 0.218	P4 0.294	P7 0.353	P2 0.448	P3 0.452	P0 0.493	P1 0.734

Table 5: The ranking order of the 2D-Vector with adaptive function ($\delta=0.66$).

Query Picture	Ranking order of the 2D-Vector with adaptive function ($\delta=0.90$). (Images shown within the same cell of a row have same relevance)											
	P0	P2	P4	P5	P6	P3	P8	P10	P7	P1	P9	P11
P0	P0	P2	P4	P5	P6	P3	P8	P10	P7	P1	P9	P11
	0.000	0.078	0.110	0.210	0.221	0.237	0.238	0.264	0.266	0.334	0.442	0.444
P2	P2	P0	P4	P3	P6	P5	P7	P10	P8	P1	P11	P9
	0.000	0.078	0.111	0.168	0.174	0.188	0.188	0.246	0.254	0.354	0.407	0.413
P4	P4	P5	P0	P2	P6	P8	P10	P7	P3	P9	P11	P1
	0.000	0.101	0.110	0.111	0.130	0.143	0.153	0.249	0.268	0.332	0.337	0.442
P6	P6	P5	P10	P4	P2	P7	P8	P0	P11	P9	P3	P1
	0.000	0.071	0.111	0.130	0.174	0.186	0.188	0.221	0.234	0.244	0.258	0.527
P8	P8	P10	P5	P4	P6	P0	P2	P9	P11	P7	P3	P1
	0.000	0.107	0.116	0.143	0.188	0.238	0.254	0.262	0.286	0.363	0.404	0.568
P10	P10	P5	P8	P6	P4	P9	P11	P2	P0	P7	P3	P1
	0.000	0.058	0.107	0.111	0.153	0.181	0.193	0.246	0.264	0.296	0.364	0.594

Table 6: The ranking order of the 2D-Vector with adaptive function ($\delta=0.90$).

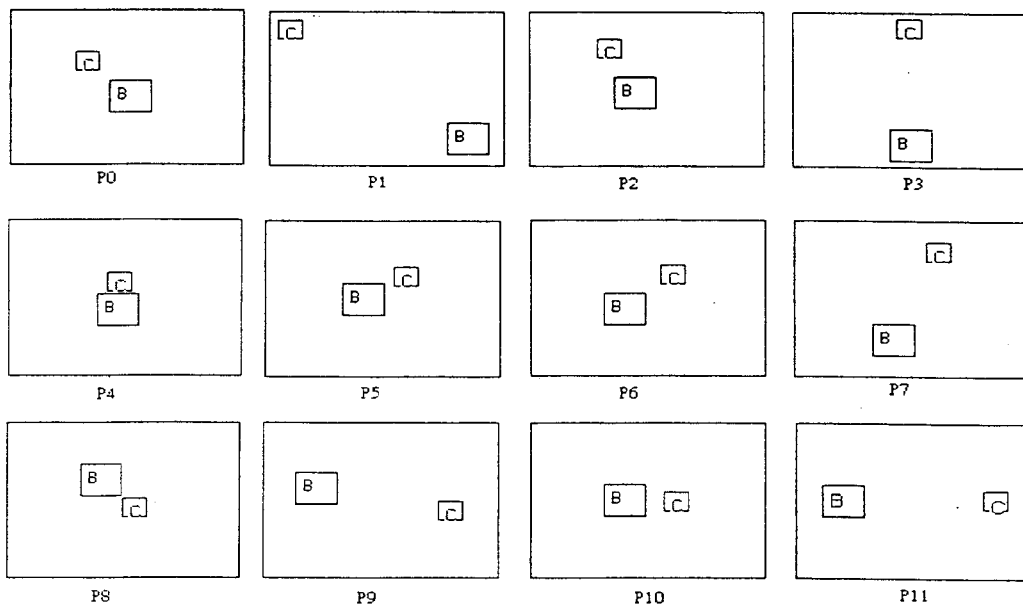


Figure 5: Test images for evaluating distance-similarity algorithm.

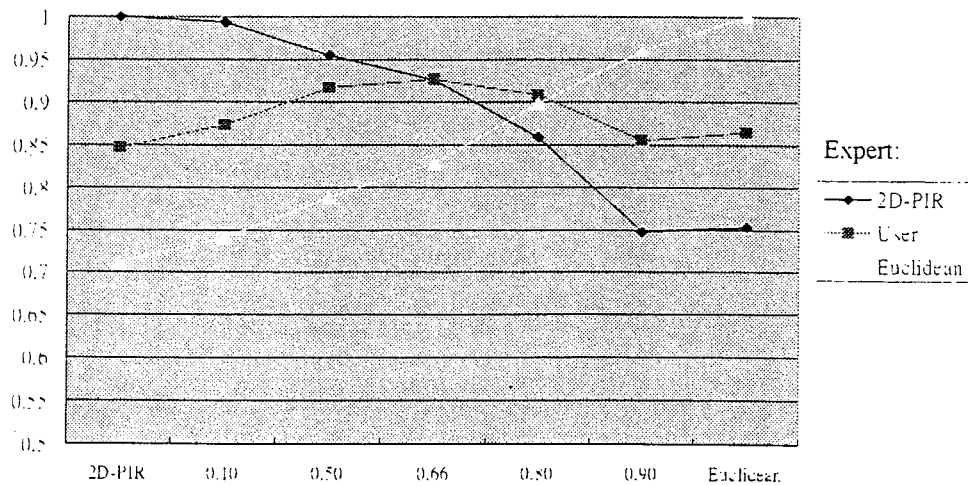


Figure 6: The distribution of average R_{norm} value.