INDEXING FOR RETRIEVAL BY SIMILARITY

Given a large set of objects (records), selecting a small subset that meets certain criteria is the central problem in database. For records whose values are alphanumeric, B-trees are used for searching. There, the query criterion has to be matched exactly. In MM objects, the objects that match the query approximately if not only exactly has also to be retrieved. To retrieve such objects in spatial database, special index structures shall have to be used. Before that we need to know what exactly similarity is. The idea is often subjective and user specific. An example is a query fingerprint image from DB storing fingerprint images. This is an example of similar picture data. Similarly, there may be similar audio data. B-trees, B+ trees, R-trees do not suffise, special index structures as to k-d trees, quad-trees are used. Mathematically, we may define similarity in terms of a transformation function with an associated transformation cost. The choice of language used to describe the transformation language determines the notion of similarity in a particular application. This transformation language applies to a pattern description language in which the multimedia object themselves are described.

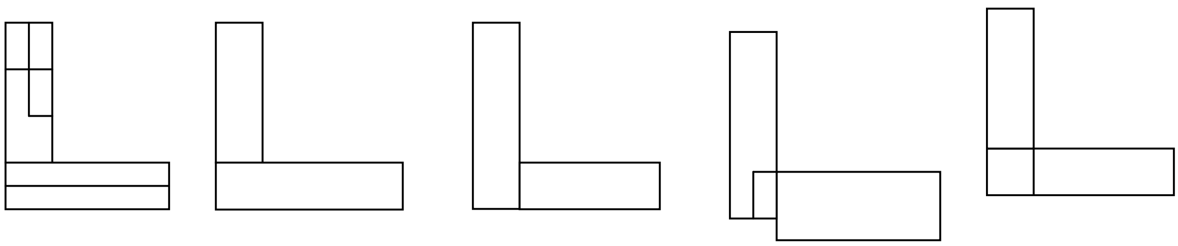
One solution techniques is to obtain an approximate "feature vector" for each object. Each MM object is mapped to a point in an attribute space. This mapping is carefully selected so that no two similar objects (with a low cost of transformation from one to the other) can be mapped to distant points.
Now, given a query point, it can be expanded to a query region of approximate size. A multi-dimension index structure can be used to retrieve objects corresponding to data points in the query region.

Shape matching:

Shape matching is an important image processing operation, most techniques of which are model driven in pattern recognition - that given a shape to be matched, it has to be compared individually against each shape in the DB or at least against a large number of clusters of features. We shall discuss data driven technique. One value of measure is the "area difference". That is two shapes are similar if the error area (where the two shapes do not match) is small when one are is placed "on top" of the other. In digital domain, we obtain a pixel wise X-OR of the two shapes and pronounce the two shapes similar if the number of pixels ON in the result is small.

RECTANGULAR SHAPE COVERS

We shall consider rectilinear shapes in two dimensions. There are two types of rectangular covers: additive and general. The former is obtained as union of several rectangles and the latter permits both addition and subtraction of rectangles with subtraction treated as a pixel-wise set difference.
The following figure shows additive rectangular covers for L shape.



(a) (b) (c) (d) (e)

Neither the additive nor the general rectangular cover for a given shape is unique. The figure shows some ways in which L-shape object can be covered additively. In the covers, we shall not permit rectangles more than the minimum required to cover the shape up. Clearly, (a), (d), (e), are disallowed. Which of the figures (b) or (c) will be chosen depends on the requirement. (b) is selected if

horizontal arm is added first & the latter if vertical arm is added first.
The features of an object may be described sequentially so that most important features are described first and any truncation of sequence is "good" approximation of the shape. A description comprises of a sequence of units. Error after first unit of description, after second unit of description and so on is accumulated till complete description is obtained. Thus, error in last stages is counted many times as compared to those in first stages or units. So error can be minimized early through best sequential description of the shape.
This is called "cumulative error criterion". To obtain an approximate description, a sequential description is obtained and truncated, leaving out less essential features of the image which is likely to have less error. Specific algorithm is not necessary. Each shape in DB is made of ordered sets of rectangles and the shape is described by relative position of rectangles.

STORAGE STRUCTURE

For each rectangle, lower left and upper right corner are identified, called L & U and can be represented by a pair of (X,Y) co-ordinates in an appropriate co-ordinate system. Thus, a set of K-rectangles can be represented by a set of 4K co-ordinates (k rectangle times 2 corners, each time 2 co-ordinates per corner). To aid retrievals, rather than storing the co-ordinates directly, transformations are applied to them. Rather than storing L,U points directly for each triangle, distinct position and size values of rectangles are obtained. The position of the rectangle is given in terms of the mean of the L & U corner points i.e. the point ($x_L + x_U/2$, $y_L + y_U/2$). x_L: X co-ordinate of L corner point. Size of the rectangle is the difference between L & U corner points i.e. ($x_U - x_L$, $y_U - y_L$). We stll have 4 values or 2 pairs of numbers for each rectangle. However, after transformation, they represent the position and size of rectangle rather than the corner points. The position of the first rectangle is used to normalize the positions of the other rectangles. Thus, the centre of the first rectangle is placed at the origin and all co-ordinates are taken w.r.t. that. The transformation is represented by a shift which is a pair of constraints to be subtracted from all X co-ordinates and all Y co-ordinates respectively. Also, as the position and size of the first rectangle is used to normalize the position and size respectively of the other rectangles. The normalization of size is referred to as the scaling. It is given as a pair of constants in the X & Y dimensions of the other rectangles considering the size of the first rectangle as (1,1).

A shape, described by a set of K rectangles can be stored as a pair of shift factors for the X & Y dimensions, a scale factor, a distortion factor, all of which are stored as part of first rectangle & as pair of X & Y co-ordinates for the centre point and a pair of X & Y size values for each of remaining K-1 rectangles after shifting and scaling. The value of the K depends on the number of rectangles required to describe a shape which can be very large. However, most of the shape information shall be given by the first few rectangles. Moreover, a similar match and not exact match is needed to match the query and retrieve from DB. So, an index on small number of K-rectangles is enough to get the information for similarity match. In this study, a point has been converted to a set of co-ordinates in 4K-dimensional space. We can now use any multi-dimensional point indexing methods like grid files and K-d trees, K-D-B trees etc.

MMDB QUERIES

There are four types of MMDB queries:

(i) Full match: or a given query shape, a full match is a DB shape that has exact shape in exact position. For ex, when we need to reproduce a particular image from a set of images, to perform full match, a query shape is transformed to a query point represented by K-rectangles & then applying the scale, distortion, and shift factors, we may use this point as a key point in the index search.

(ii) Match with shift: When we are concerned with the shape of an image, we do not care much about the position. This method is used to retrieve similar shapes irrespective of their positions. In the co-ordinate system. It is done by transforming the query shape to a point and then "throwing away" the two shift factor co-ordinates of the point, making the query region infinite around the point allowing any value of shift factor. The query region may be used as the key in index search which retrieve as data matching all points except two shift factors.

(iii) Match with uniform scaling: Here, the size of the shape also not considered along with the position co-ordinate. The scale factor is also thrown away along with the shift factor and then retrieval is performed.

(iv) Match with independent scaling: Rather than uniform scaling along X & Y axis, independent scaling on either axis is performed. This is for pictures taken at an angle to the shape.

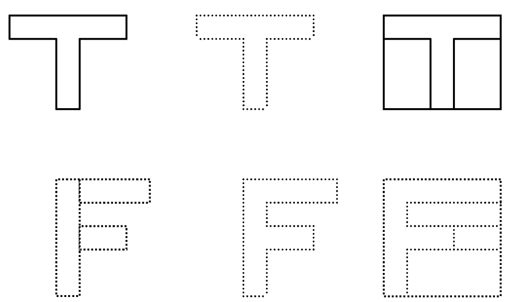
APPROXIMATE MATCH WITH APPROXIMATION PARAMETERS

We have already discussed approximate match in terms of similarity matches. How the approximation is done is to obtain objects of similar shapes is by retrieving all objects whose shape descriptions have rectangle with similar position and size, if not identical with query shape position and size.

This is done by "blurring" the query point by specifying a range along the attribute axis, corresponding to some flexibility w.r.t. exact values for position and size for each rectangle. The extent of blurring is determined for each attribute axis independently by using appropriate parameters. The larger the amount of "blur", the weaker is the search criterion & larger is the set of objects selected to be "similar" to query shape. We have seen shape descriptions to be sequential instead of arbitrary. A better way of approximation is to use the first few rectangles of the shape that carry maximum information about shape. Since shape matching is not exact, similar shapes may differ only in last few rectangles. The number of rectangles K to be used in search has to be carefully chosen. Too few rectangles may produce dissimilar shapes in the result of the query, different from original query shape. Too many rectangles are not used as it is not essential. One heuristic rule is to truncate the description when error area becomes reasonably small fraction of the total area. Another heuristic is to is to truncate the description when size of error of next rectangle becomes a small fraction of the total area.

MULTIPLE REPRESENTATION:

One problem with similarity retrieval is that fairly similar shapes may have different optimal (sequential) descriptions.



Two shapes T & F are shown. As the relative sizes of the points of the parts change, the sequential description for the general rectangular cover also changes. As the sequential or optimal description changes, it becomes difficult for human eye to distinguish the change even though their optimal descriptions are completely different. Mathematically different descriptions may seem similar to human senses. This problem occurs not only for general covers but for additional covers too. Almost same shapes will map o different points in the attribute space over which index is applied as the choices of sequential descriptions occur. One solution is to retain two or more sequential descriptions that give similar shapes perceptible to human descriptions eye. But it has the disadvantage of increasing the size of DB. A better solution is to obtain multiple "good" sequential descriptions of the given query shape & then perform a query on each of them. In this solution, query time increases but DB & Index structure in not increased.

A DATA ACCESS STUCTURE FOR FILTERING DISTANCE QUERIES IN IMAGE RETRIEVAL

In the structure of image information system, there are three stages in image processing.

1. Image analysis & pattern recognition
2. Image structuring & image understanding
3. Spatial reasoning & image retrieval

The first step concerns interpretation of an image content to recognize a set of objects from raw images. In the second stage, some image knowledge structures are constructed so that spatial reasoning & image retrieval can b supported. Such structures can be semantic networks, topological structures like directed graphs showing spatial relationships or spatial indexes for recognized image objects. Final stage represents the interactions between user applications and the image DB systems, some of these applications are oriented to spatial reasoning, others to image retrieval. The wide range of applications of image DBs produces different requirements like:

1. The image DBs consists of set of images that describe objects embedded in some space. The scenario implies that spatial links between images & space but also between images

themselves are to be represented in the model. ***Spatial reasoning*** is used in querying such DB. Ex: GIS applications.

2. The image DB consists of set of images that describe different instances of the same object types, these instances can belong to a temporal series or describe a set of distance physical objects. Image retrieval with similarity predicates, is used for such cases. Picture management (with person faces) especially medical picture management is an example of this type.

Case 1 emphasizes on image objects.

Case 2 emphasizes image as a whole as target for query where a set of images could be result of the query. Two most important predicate of image retrieval are:

1. Spatial predicate
2. Similarity predicate.

In both these predicates, spatial relationship plays an important part. In first one, spatial relationship is derived by embedding all the images in same reference space, in the second case, similarity predicates can be based on spatial relationships between image objects contained in the same image. Two objects are similar if some spatial relationship exists among their objects.

Because spatial relationships are relevant in image DBs, special data access structures called SAM i.e. Spatial Access methods used to optimize the section of image objects or selection of images in query involving spatial predicates. Traditional query methods are not sufficient as volume of data in image DB is much larger compared to traditional DB, query is richer, and at physical levels, there are raw, unstructured images. Several SAMs are there but our discussion shall be limited to queries involving distance concept between two image objects also called metric relationships. The basic components of Sam are RDBMS, IP (Image Processing), in one module and hi-level of QI (Query Interpreter) in second module, both integrated by some linkage pointers between the two parts of information.

SPATIAL ACCESS METHODS AND IMAGE RETRIEVAL

QUERY PROCESSOR

Images are huge data objects & access to secondary storage to retrieve such huge objects is more time consuming that traditional DBs. In order to reduce the amount of data that has been loaded in main memory to process an image, different levels of data structure are constructed above main memory so that during query processing, the number of images to be processed in main memory is reduced. A set of image objects represents lowest level of hierarchical structure that describes content of image in DB. Image objects can be represented by a single point giving location of image, through vector representation of boundary, or thru Minimum Boundary rectangle (MBR). A higher level data structure represents relationships between the image objects& for spatial relationships, SAM is used. Using SAM, spatial queries are processes accordingly.

1. An initial filter phase: Uses Spatial Access method to identify a set of candidates which could be contained in the query result.
2. A successive refinement phase: I uses algorithms of computational geometry which implements the query predicate to set f candidates obtained in step 1. to calculate the final result.

In image retrieval, spatial query processing is useful as:

1. When emphasis is on image objects, it reduces set of image objects to be loaded in the main memory for query processing. If query execution requires to load whole image that contains object, the images

considered are those with at least one candidate image object.

2. When emphasis is on one single image, it reduces the set of images to be loaded in the main memory for query processing. Implementation of spatial predicate based on computational geometry is much more time expensive than equality and range predicates in traditional DBs reducing number of images to be processed that can reduce the total time of processing the query.

In both cases, performance increases.

IMAGE OBJECTS - SPATIAL PREDCATES

Image query language use s set of spatial predicates on image objects as a part of image retrieval. The features of an image object may vary in accordance with interpretation criteria of pattern recognition, so it is difficult to fix the properties of an image object. However, an image object must have a shape and location on an image somewhere. We limit the address space to Euclidean space E^2.

An image object is a planar shape embedded in the Euclidean plane E^2 which can be represented as a closed set of points. The set of all image objects is IO.

$IO = \{ g \mid g \subset E^2 \ \wedge g \text{ I closed}\}$

Two geometric functions form IO to E^2 can be defined

Boundary: IO → E^2

Boundary (g) = δg where $\delta g \leq g$.

It returns the portion of input image object that can be considered as the frontier of the object itself. Any boundary condition can be used if it satisfies the condition: $\delta g \leq g$.

In some image applications, boundary of an image object could be for example, a buffer region around the object itself.

Interior IO → E^2

Interior (g) = g^0 where g^0 = g- δg

It returns the portion of the input image object that is not the boundary of the object itself. Spatial relationships in the Euclidean plane can be divided to two groups such as:

1. Topological relationships which are completely independent of distance concept & involve close image objects which have inferences among them.
2. Distance relationships: When the function distance is introduced in embedded space, it represents how far two images objects are from each other.

Let us take up certain spatial quire having distance relationships. For ex:
Q1. Select the image objects representing the blocks in the town “X” where people who have Z syndrome live.
Q2. Select the image objects representing villages within 10 miles from a toxic waste dump D
OR
The queries can produce a set of images. For ex:
Q3. Select all the images representing a flooding where the water reached the main hospital of town “X”.
OR
The queries can produce a set of alphanumeric data. For ex:
Q4. Select names and addresses of all patients who live within 1 mile from river “Y” & has “Z” syndrome.

The queries may contain alphanumeric predicates along with spatial predicates. The alphanumeric part is processed by conventional RDBVMS which produces a set of candidate images/objects. The final result is obtained by the Image processor considering the spatial part of the predicate.

ACCESS DATA STRUCTURES FOR SPATIAL DATA

This is a combination of several techniques for traditional DBs & access structures for spatial data. These techniques are regular grid with locational key, clustering technique for spatial structures in R+ trees and extensible hashing.

REGULAR GRID WITH LOCATIONAL KEYS

This is based on regular, recursive sub-division of space. The pace is initially partitioned to 4 sub-quadrants called MW. NE, SW, & SE. To each subquadrant, the keys 00,01,10,11 are assigned. The plane is then recursively partitioned until a desired, detailed level is reached. The sub-quadrants obtained in the last step are called cells. Let K be the key of a given quadrant called quad(K); decomposition of which to subquadrants yield the following 4 subquadrant keys:

	K.00	Key of NW SQ of Quad (K)
	K.01	NE
K $\Rightarrow$	K.10	SW
	K.11	SE

Where. Operator is string concatenated. Let us take an example where Quad(01) generates 4 keys for SQ 0100, 0101, 0110, 0111.

If l is level of recursion, 2Xl is the length of key. Recursion produces regular grid.