Similarity queries can be classified into following two categories as shown in Fig. 4.10.1.

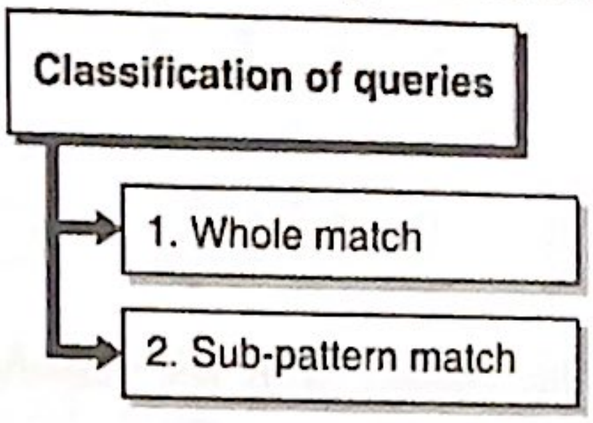


Fig. 4.10.1 : Classification of queries

1. Whole match

- If the collection of N objects O1, O2, ..., ON and a query Q is given we have to find those objects that are within distance $\in$ from Q.
- In this match query and objects are of the same type.
- For example is the objects are 512×512 gray-sale images then only we can have query of whole match type.

2. Sub-pattern match

- The query is allowed to specify only part of the object.
- Given N data objects O1, O2, ..., ON and a query (sub) object Q and a tolerances $\in$, we want to identify the parts of the object that match the query.
- For example if the objects are 512×512 gray-sale images like medical x-rays then the query could be 48×48 sub pattern e.g. a typical x-ray of tumour.

Additional types of queries includes

- Nearest neighbours queries :

  E.g. find the five most similar stocks to IBM's stock.
- All pair queries or spatial joins :

  E.g. Report all the pairs of stocks that are within distance $\in$ from each other.

  The ideal method should fulfill the following requirement for all the types of queries :
- As sequential scanning and distance calculation with each and every object will be too slow for large databases so it should be fast.
- It should be correct that means it should return all the qualifying objects, without missing any :

  'False alarmis' are acceptable as they can be discarded easily through a post processing step but try to keep the number low so that the total response time is minimized.
- It should require a small space overhead.
- The method should be dynamic. If method is dynamic then it is easy to insert, delete and update the objects.

4.10.2 Spatial Access Methods

- In 'GEMINI' approaches if feature extraction functions are used to map objects into points in f-dimensional space.
- To accelerate the search this we can use highly fine-tuned database spatial access methods.

- The spatial access methods form three classes :
  1. R* - trees and the R - tree family
  2. Linear quad trees
  3. Grid - files
- Several of these methods explode exponentially with the dimensionality eventually reducing to sequential scanning.
- For linear quad trees, the effort is proportional to the hyper surface of the grows exponentially with the dimensionality.
- Grid files also face the similar problem as they a directory that grows exponentially with the dimensionality.
- Grid files also face the similar problem as they require a directory that grows exponentially with the dimensionality.
- The R-tree based method seems to be more robust for higher dimensions, provided that the famous of the R-tree node remain greater than two.
- As one of the typical representatives of spatial access method in following section we discuss the R-tree method and its variants.

R-trees

- Represents a spatial object by its Minimum Bounding Rectangle (MBR).
- Parent nodes are formed by grouping data rectangles and parent nodes are recursively grouped to form grandparent nodes. And eventually a tree is formed.
- The MBR of parent node completely contains the MBR's of its children. And MBR are allowed to overlap. Modes of the tree correspond to disk space.
- Disk pages or disk blocks are consecutive byte positions on the surface of the disk that are typically fetched with one disk access.
- The goal of the insertion split and deletion routines is to give trees that will have good clustering with few light parents MBR.
- Fig. 4.10.2 illustrates data rectangles organized in an R-tree with fanout 3.

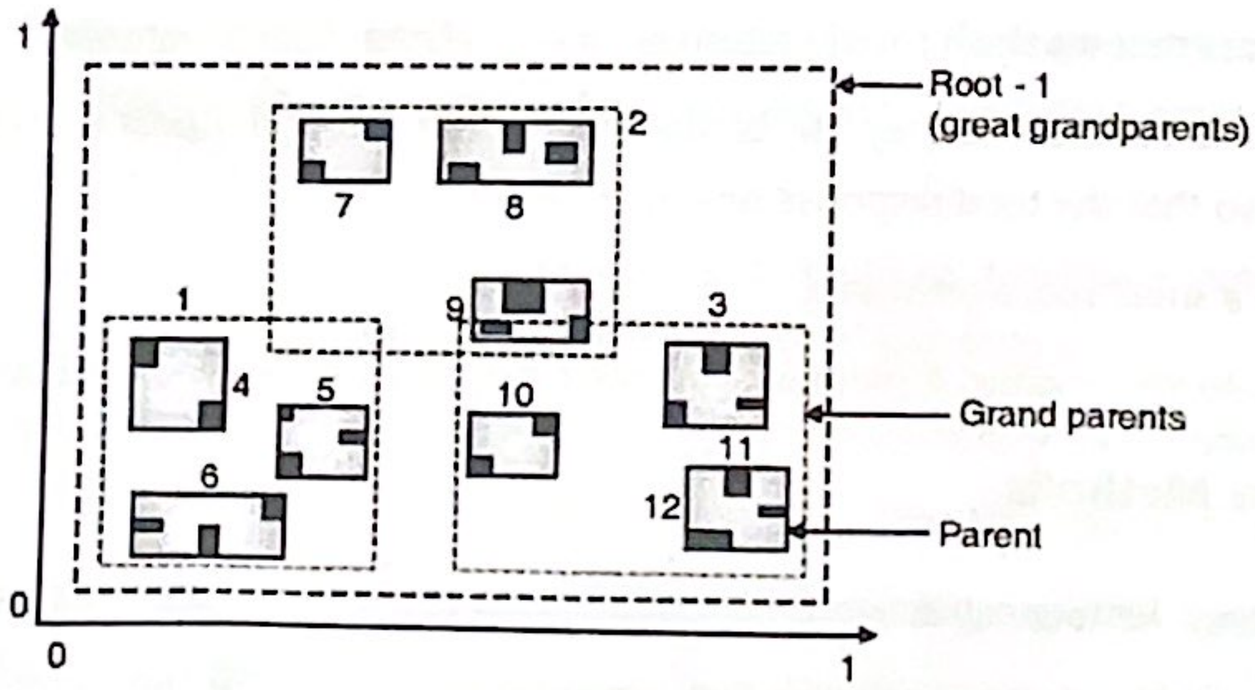


Fig. 4.10.2 : Organization of R-tree with fanout 3

- Fig. 4.10.3 shows the file structure for the same R-tree where nodes corresponds to disk pages.

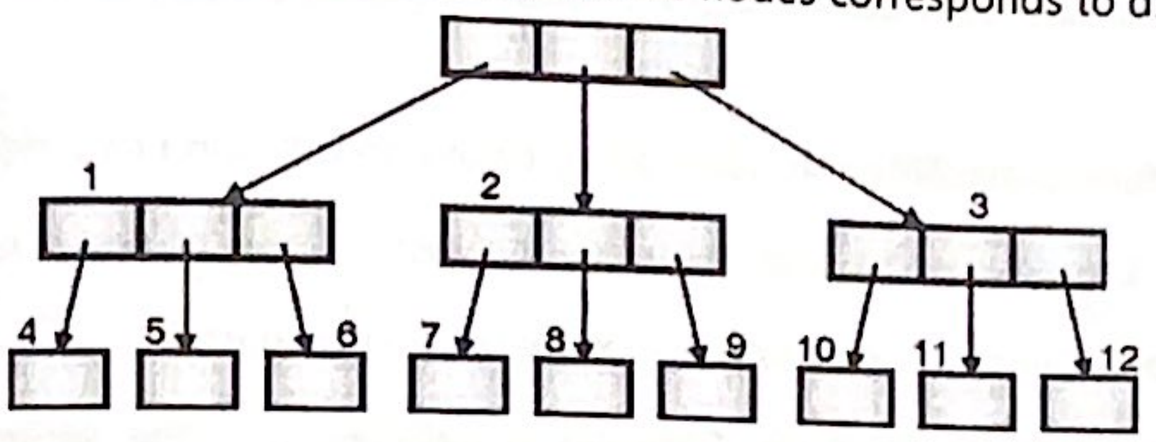


Fig. 4.10.3 : The file structure for the R-tree with fanout 3

- A range query specifies a region of interest requiring all the data regions that interest it.
- To answer this type of query perform following steps :
  1. First retrieve a superset of the qualifying data region.
  2. Compute the MBR of the data region.
  3. Then recursively descend the R-tree excluding the branches whose MBR do not intersect the query MBR.
- Thus the R-tree will give you quickly the data regions whose MBR intersects, the MBR of the query region.
- Then the retrieved data regions will be further examined for intersection with the query region.

4.11 Generic Multimedia Indexing Approach

University Questions

Q. Explain the generic multimedia indexing approach. **SPPU : May 12, Dec. 12, Dec. 16, 8 Marks**

Q. Explain GEMINI approach for multimedia IR. **SPPU : May 13, May 16, May 19, 8/9 Marks**

Q. Write short notes on : Generic multimedia indexing. **SPPU : Dec. 14, 4 Marks**

For whole match queries the problem is defined as follows :

- We have collection of N objects : O_1, O_2, ..., O_N
- The distance or dissimilarity between two objects (O_i, O_j) is given by the function D (O_i, O_j). This function is implemented as program.
- The user has to specify the query object Q and a tolerance $\in$.
- Here we have to find the objects in the collection of N objects O_1, O_2, ..., $O_N$that are within distance $\in$ from the query object.
- One solution to find such objects is to apply sequential scanningi.e. for each and every object O_i where $1 \leq i \leq N$ we have to compute its distance from Q and report the objects with distance $D(Q,O_i) \leq \in$.
- But the sequential scanning may be slow because of following reasons :
  1. The computation of distance might be expensive.

     For example the editing distance in DNA strings requires a dynamic programming algorithm which grows (in the range of hundreds or thousands) like the product of the string lengths.
  2. The data base size H may be huge.

- In order to avoid above mentioned disadvantages of sequential scanning we need faster alternative and that is nothing but GEMINI.
- The GEMINI i.e. GEneric Multimedia INdexIng approach is based on following two ideas :
  1. 'Quick and dirty' test is performed to discard quickly the vast majority of non qualifying objects.
  2. To achieve fasters than sequential searching use of spatial access method.
- We will illustrate the case with an example. Consider a database of time series, such as yearly stock price movements with one price per day.
- Assume that the distance function between two such series S and Q is the Euclidian distance and given as :

$$D(S, Q) = \left(\sum_{i=1} (S[i] - Q[i]^2)^{1/2} \right)$$

Where,

S [i] : The value of stocks on the i^{th} day.

- In this example computing the distance of two stocks will take 365 subtractions and 365 squarings.
- In case of quick and dirty test we have to characterize a sequence with a single number.
- This characterization will help us to discard many non-qualifying sequences.
- And such a number could be the average stock price over the year, if we consider the above example.
- If two stocks differ in their average by a large margin then it is impossible that they will be similar.
- But the converse is not true and because of that we have the false alarms i.e. non-qualifying objects.
- With good feature (numbers that contain some information about a sequence is referred as features) we will have a quick test, which will discard many stocks with a single numerical comparison for each sequence.
- Using two or more features might be better if using one feature is good, because it will reduce the number of false alarms. But it will make quick and dirty test a bit more elaborate and expensive and because of that cost will be increased.
- In stock price example one feature is Eudidian distance and additional features can be standard deviation, Discrete Fourier Transform (DFT) coefficients.
- So finally we can map each object joint a point in f-dimensional space by using f features for each of objects. And this mapping is referred as F( ). (Here F stands for feature).
- Mapping F ( ) is defined as - the mapping of objects to f-dimensional points i.e. F (0) is the f - D point that corresponds to object 0.
- This mapping helps to improve the second drawback of sequential scanning.
- We can do this by organizing f - D points into a spatial access method and clustering them in a hierarchical structure like the R* tree.
- So depending upon the query, we can prune out the large portion of the database that are not promising by exploiting the R* tree.
- Fig. 4.11.1 illustrates the basic idea. It shows a database of sequences of S_1, S_2, ..., S_N.
- It shows each sequence is mapped to point in feature space.

- And a query with tolerance $\in$ becomes a sphere of radius $\in$.

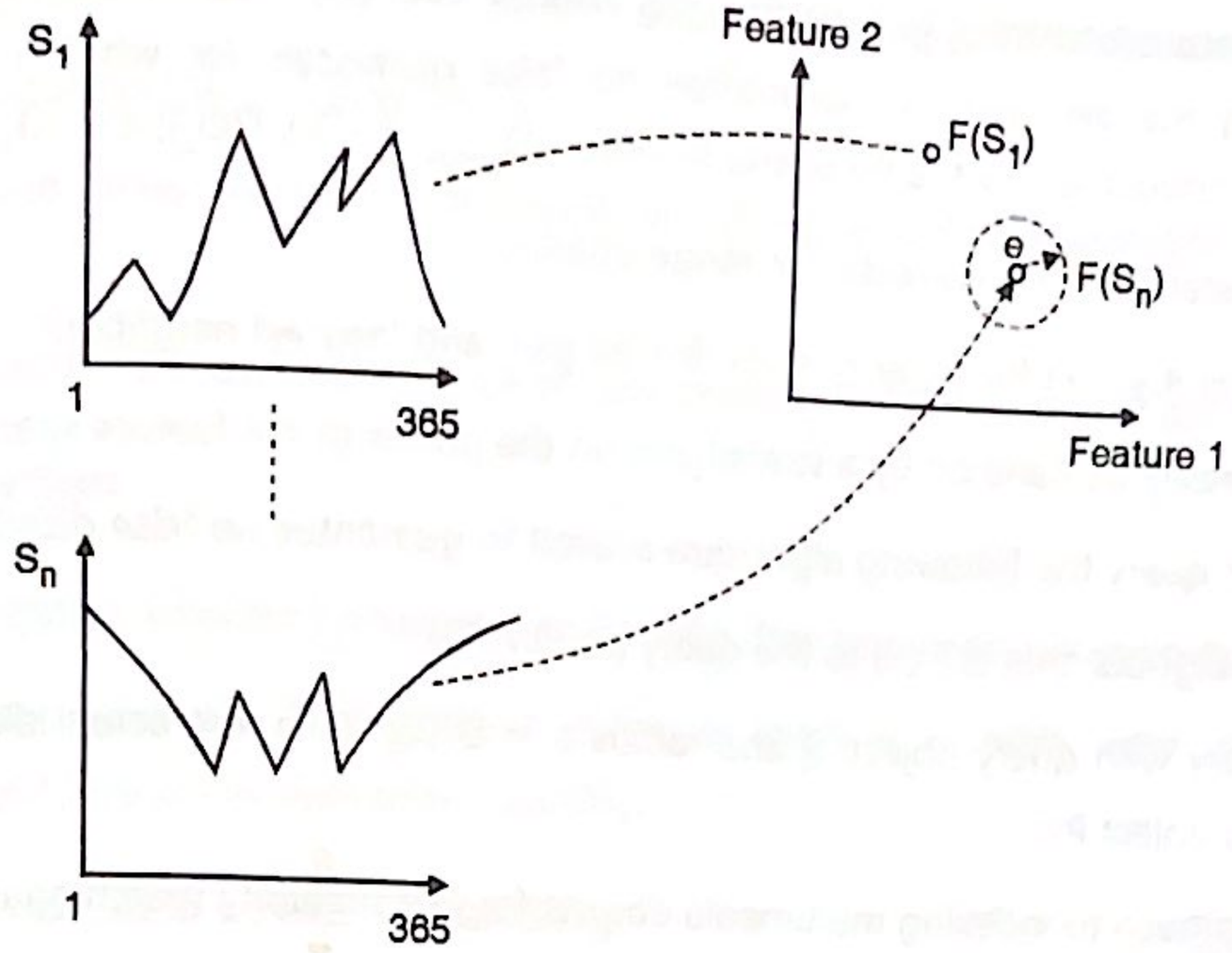


Fig. 4.11.1 : Illustration of basic idea

- Here the objects are mapped into 2D points. If we consider the stock price example then objects are time series that are 365 points long are mapped into 2D points and points are using the average and the standard deviation as features.
- Consider the 'whole match' query that requires all the objects that are similar to $S_N$within tolerance $\in$ then this query becomes an f - D sphere in feature space. Centred on the image F (S_N) of S_N.
- Then the search algorithm for a 'whole match' query is as follows :
  1. In feature space map a query object Q into a point F (Q).
  2. Retrieve all the points which are within the desired tolerance by using a spatial access method.
  3. Then retrieve the corresponding objects and then compute their actual distance from Q and discard the false alarms.
- So the method has the potential to relieve both problems of the sequential scanning and result in faster searchers.
- But we have to be careful at the time of mapping F () from objects to F - D points such that the distance should not get distorted.
- Let D () be the distance function of two objects and $D_{feature}$ ( ) be the Eudidian distance of the corresponding feature vector, then the mapping should preserve the distances exactly, in which case the spatial access method will have neither false alarms nor false dismissals.
- But requiring perfect distance preserve distance preservation might be difficult. Because for example, it is not obvious which features we have to use to match the editing distance between two DNA strings.
- Even if the features are obvious, there might be practical problems, for example in the stock price example, we could treat every sequence as a 365 - dimensional vector, although in theory a spatial access method can support an arbitrary number of dimensions and in practice they all suffer from the 'dimensionality curse'.
- If the distance in feature space matches or underestimates the distances between two objects then we can guarantee that there will be no false dismissals.

- If $O_1 \times O_2$ be two objects (e. g. same-length sequence) with distance function D ( ) (e.g. the Euclidean distance) and F (O_1), F(O_2) be their feature vectors (e.g. their first few Fourier Coefficients) with distance function $D_{feature}$ ( ) (e.g. the Euclidean distance) the we have "To guarantee no false dismissals for whole-match queries the feature extraction function F ( ) should satisfy the following formula : $D_{feature}$ (F (O_1), F(O_2)) ≤ D (O_1, O_2)
- Lower-bounding the distance works correctly for range queries.
- Now question is whether it works for other queries like 'all pair' and 'nearest neighbour'.
- An 'all pair' query can easily be handled by a spatial join on the points of the feature space.
- For 'nearest neighbour' query the following algorithm is used to guarantee no false dismissals :
  1. Find the nearest neighbor that is F (P) to the query point F (Q).
  2. Issue a range query with query object Q and radius ϵ = D (Q, P) i.e. the actual distance between the query object Q and data object P.
- Finally the GEMINI approach to indexing multimedia objects for fast similarity searching is as follows :
  1. Find the distance function D ( ) between two objects.
  2. To provide a 'quick and dirty' test find out one or more numerical feature extraction functions.
  3. To guarantee correctness, prove that the distance in feature space lower bounds the actual distance D ( ).
  4. To store and retrieve the f-D feature vectors use a spatial access method (e. g. an R-tree).
- To find the distance function D ( ) between two objects involves a domain expert.
- The methodology focuses on the speed of search but the quality of the results is completely depends on the distance function that the expert will provide.
- Finding numerical feature extraction function will require intuition and imagination.
- It starts by trying to answer the question called as 'feature-extraction' question. The feature extracting questions can be of - "If we are allowed to use only a numerical feature to describe each data object, what should this feature be ?"
- The successful answer to the above question will provide two goals :
  1. They should facilitate step 3 i.e. the distance lower bounding.
  2. They should capture most of the characteristics of the objects.
- Thus, GEMINI will return exactly the same quality of output that would be returned by a sequential scanning of the database.
- And the GEMINI is faster than sequential scanning.

4.12 One-Dimensional Time Series

University Question

Q. Explain one dimensional time series. **SPPU : May 14, 8 Marks**

- We will give case studies of steps 2 and 3 of the GEMINI algorithm using
  1. One dimensional time series
  2. Two dimensional color images

- For each case study, first we will describe the objects and the distance function, and then we will show how to apply lower-bounding lemma and finally give experimental results on real or realistic data.
- In this section we will discuss the 1-D time series.
- In one dimensional time series, we have to find series which are similar to a desirable series from the collection of time series.
- For example "In a collection of yearly stock price of movements. Find the ones that are similar to IBM".

4.12.1 Distance Function

- The first step in the GEMINI algorithm is we have to find the distance measure between tow time series.
- And a typical distance functions the Euclidean distance which is of reply used in financial and forecasting application. Additional functions include time - warping.

4.12.2 Feature Extraction and Lower -Bounding

- After deciding the distance function D ( ) between two objects i.e. Euclidean distance function. Next we have to find some features that can lower bound it.
- We have to find out the set of features that will satisfy the two requirements :
  1. They should lower bound the distance.
  2. They should carry much information about the corresponding time series so that the false alarms are less.
- The above second requirement suggests that we use 'good' features that have much discriminatory power.

  In the stock price example a 'bad' feature is 'the first day's value' because two stocks might have similar first-day values though they may differ significantly from them.
- Even though if we use the values of all 365 days as features, this would perfectly match the actual distance but it would lead to the 'dimensionality curse' problem. So we need better features.
- In the second step of GEMINI algorithm, we have to ask feature - extracting question of the form If we are allowed to use only one feature from each sequence, what would this feature be ?. And answer to this type of question is the average.
- By the same token additional features could be the average of the first half, of the second half of the first quarter etc.
- That means we could use the coefficient of the Discrete Fourier Transform (DFT).
- For a signal $\vec{x} = [x_i]$, $i = 0, \ldots n-1$, and x_F denote the n - point DFT coefficient at the F^{th} frequency ($F = 0, 1, \ldots n-1$).
- In the third step of GEMINI algorithm we have to show that the distance in feature space lower-bounds the actual distance.
- This solution is provided by Parsevals theorem. It states that, the DFT preserves the energy of a signal as well as the distances between two signals.

$$D(\vec{x}, \vec{y}) = D(\vec{X}, \vec{Y})$$

Where, $\vec{X} \times \vec{Y}$ are Fourier transforms of $\vec{x}$ and $\vec{y}$ resp.

- Thus by keeping the first f ($f \leq n$) coefficients of the DFT as the features we can lower bound the actual distance as :

$$D_{feature}(F(\vec{x}), F(\vec{y})) = \sum_{F=0}^{f-1} |X_F - Y_F|^2 \leq \sum_{F=0}^{n-1} |X_F - Y_F|^2$$

$$= \sum_{i=0}^{n-1} |x_i - y_i|^2$$

And finally : $D_{feature}(F(\vec{x}), F(\vec{y})) = D(\vec{x}, \vec{y})$

Thus there will be no false dismissals.

- GEMINI algorithm can be applied with and ortho normal transform. Such as Discrete Cosine Transform (DCT), the wavelet transforms because they all preserve the distance between the original and the transformed space.
- The response time will improve with the ability of the transform to concentrate the energy, the fewer the coefficients that contain most of the energy, more accurate will be the estimates for the actual distance, the fewer the false alarms and faster the response time.
- Thus, better transforms will achieve the better response times.
- The DFT concentrates the energy in the first few coefficients for a large class of signals i.e. the colored noises. These signals have a skewed energy spectrum ($O(F^{-b})$) as follows :
  1. For the value of b = 2 we will have random walks or brown noise. And such signals model successfully stock movements and exchange rates.
  2. With the value b > 2 i.e. more skewed spectrum we will have black noises. And such kind of signals models successfully, the water level of river and the rainfall patterns as they vary over time.
  3. With the b = 1 value we will have pink signals. Musical scores and other works of art, consists of pink noise. Whose energy spectrum follows $O(F^{-1})$.
- In general white noise with $O(F^0)$ energy spectrum is completely unpredictable, brown noise with $O(F^{-2})$ energy spectrum is too predictable and 'boring' and the energy the spectrum of pink noise lies in bean.
- Fig. 4.12.1 shows the movement of the exchange rate between the Swiss France and the US dollar starting 7th August 1990 to 18th April 1991 and first 3000 values out of 30,000.

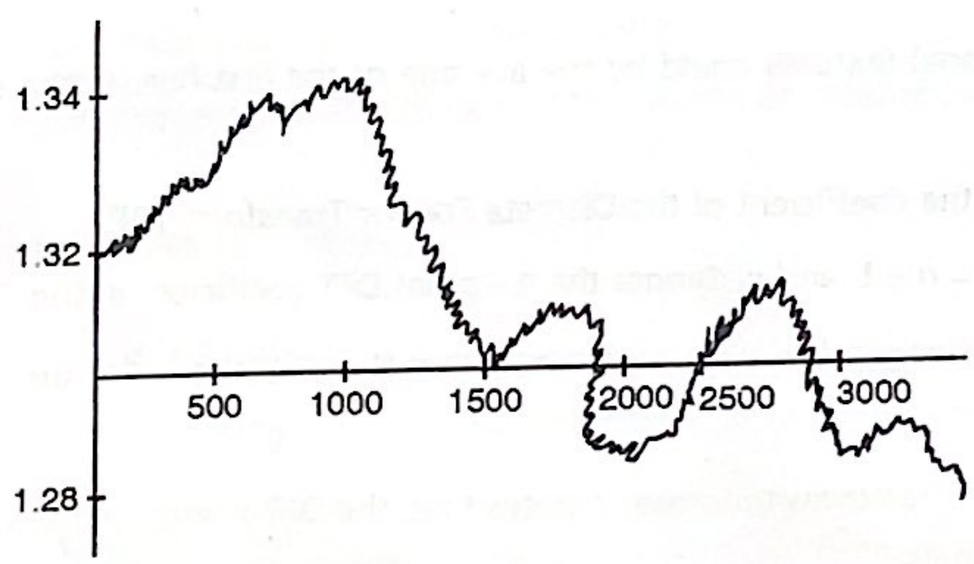


Fig. 4.12.1 : Movement of the exchange rate

- Fig. 4.12.2 shows the amplitude of the Fourier coefficients as a function of the frequency F, as well as the 1/F line in a logarithmic - logarithmic plot.

- As it is successfully modelled as a random walk, the amplitude of the Fourier coefficients follows the 1/F line.

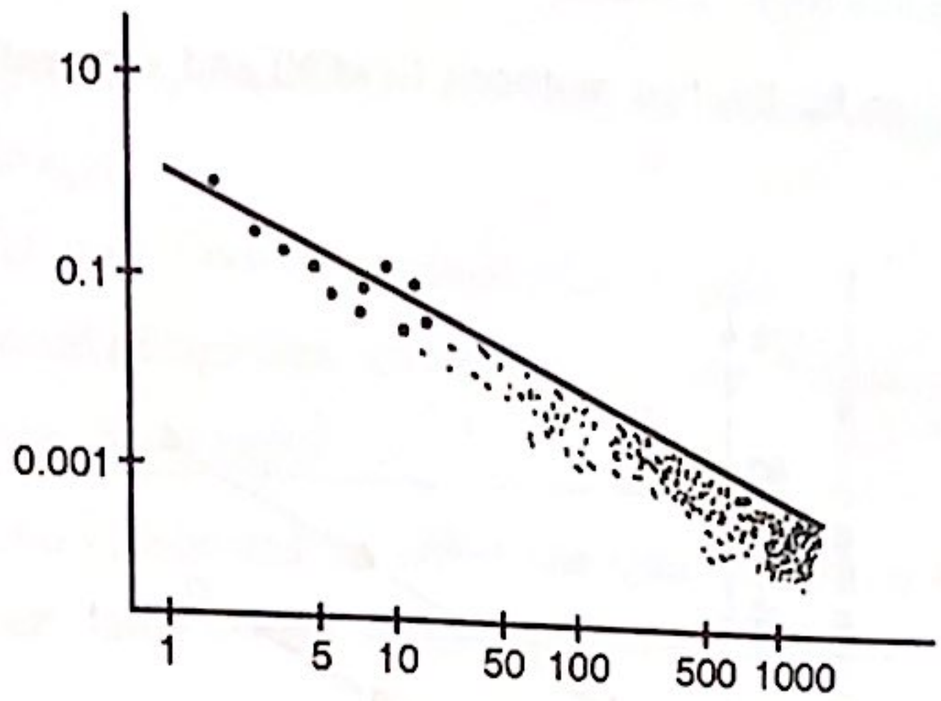


Fig. 4.12.2 : Amplitude spectrum of Fourier transform

- In addition to 1D signals, several families of real n - D signals belong to the family of colored noise, with skewed spectrum.

4.12.3 Experiments

- Performance results with the GEMINI approach on time series are collected in this section from the reference, "Efficient similarity search in sequence database, by Rakesh Agrawal, Christos Faloutsos and Aram swam."
- In that the method is compared with the sequential scanning method.

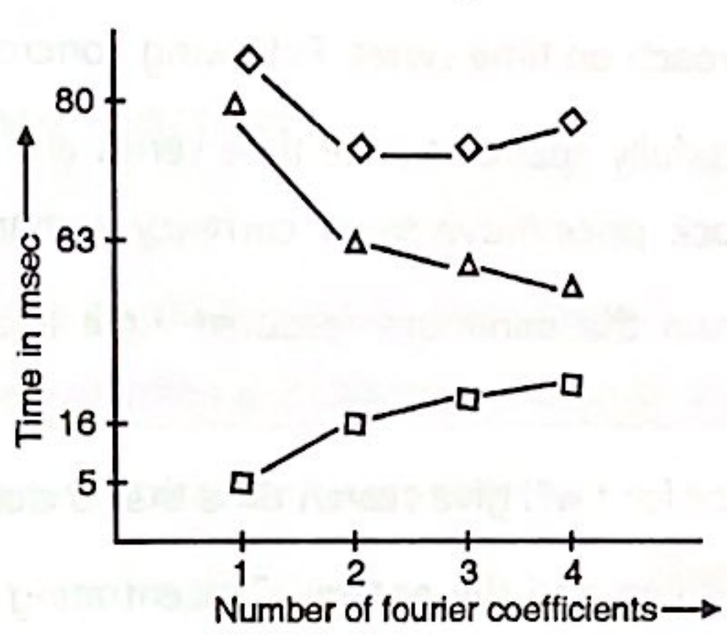


Fig. 4.12.3 : Breakup of the executiontime for range query

- The R* - tree is used for the spatial access method within GEMINI.
- The sequences were artificially generated random walks, with length n = 1024 and number N varied from 50 to 400.
- Fig. 4.12.3 shows the break-up of the response time as a function of the number f of DFT coefficient.
- In the Fig. 4.12.3,

  ◊ (Diamond) – Total time

  Δ (Triangle) – Post - processing time

  □ (Square) – R* tree time.
- If we keep more features F_1 the R* tree becomes bigger and slower but more accurate. And if it is and shorter post - processing time.

- This trade off reaches an equilibrium for f = 2 or 3. For the remaining experiments the f = 2 Fourier coefficients were kept for indexing, resulting in a four - dimensional R* tree.
- Fig. 4.12.4 shows the response time for the two methods GEMINI and sequential scan, as a function of the number of sequences H.

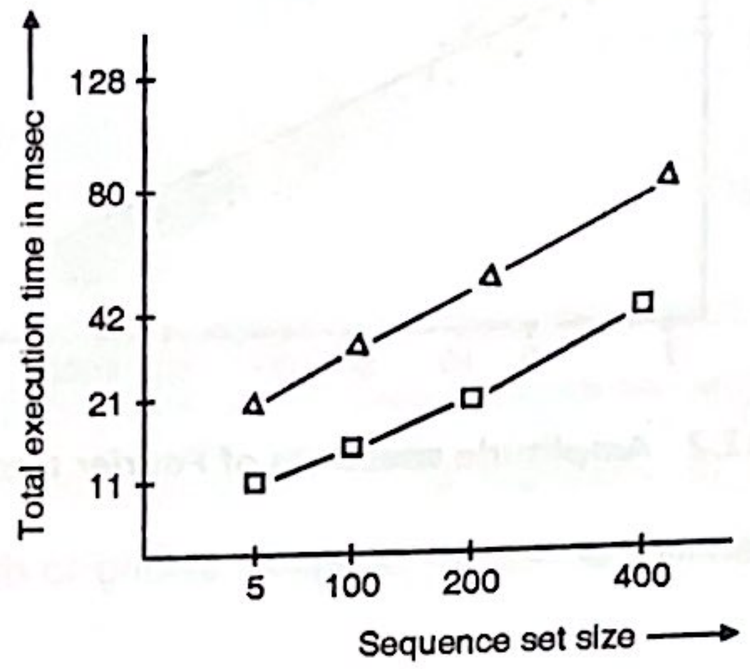


Fig. 4.12.4 : Search time per query VS number N of sequences

- In the Fig. 4.12.4 search time per query VS number N of sequences for whole match queries is shown. In the Fig. 4.12.4 the line with □ represents GEMINI approach and the line with Δ shows sequential scanning.
- And from the Fig. 4.12.4 it is clear that GEMINI approach outperforms the sequential scanning.
- From the application of GEMINI approach on time series. Following concretions are drawn :

1. GEMINI approach can be successfully applied to the time series and specifically to the ones that behave like 'colored images' for example stock price movements, currency exchange rates, water level in rivers etc.
2. For signals with skewed spectrum the minimum response time is achieved for a small number of Fourier coefficients i.e. f = 1, 2, 3.

   It means that a suboptimal choice for f will give search time that is close to the minimum.

   With the help of the lower bounding and the energy. Concentrating properties of the DFT we can avoid the 'dimensionality curse'.
3. The success in 1D series suggests that GEMINI is promising for 2D or higher dimensionality signals. If those signals also have skewed spectrum.
4. The success of JPEG images which uses DCT indicates that real images indeed have a skewed spectrum.
5. The method is extended to handle sub pattern matching for time sequences.

- Assuming that query pattern have length of at least w we pre-process every sequence of the database by allowing a sliding window of length w at each and every possible position and by extracting the features f for a given positioning of the window.
- Thus, every sequence becomes a trail in the f - dimensional feature space which can be further approximated by a set of few MBR's that cover it. Representing each sequence by a few MBR's in feature space may allow false alarms but no false dismissals.
- The same approach can be generalized for sub pattern matching in 2D signals and in general for n-dimensional vector fields.

4.13 Two-Dimensional Colour Images

University Questions

Q. Discuss the application of the GEMINI approach for Two- dimensionalcolor images. **SPPU : May 12, 8 Marks**

Q. Write short note on : 2-D color images. **SPPU : Dec. 12, 8 Marks**

Q. Explain GEMINI algorithm for indexing of two dimensional color images. **SPPU : Dec. 13, 10 Marks**

Q. Explain how image can be retrieved using image contents as the basis of retrieval. **SPPU : Dec. 13, 8 Marks**

Q. Explain how GEMINI is applied for color images. **SPPU : May 15, 8 Marks**

- GEMINI has also been applied for colour images within the QBIC (Query by Image Content) project of IBM. This project studies methods to query large online images databases using the images content as the basis of the queries.
- Examples of the content include colour, texture, shape position and dominant edges of image items and regions.
- In this section we will discuss methods on databases of still images with two main data types :
  1. Images or Scenes : A scene is a color image.
  2. Items : A item is a part of a scene.
- For example a person, a piece of outlined texture or an apple.
- Each scene has zero or more items.
- In this section we will give an overview of the indexing aspects of QBIC and specifically the distance functions and the application of the GEMINI approach.

Feature of image can be extracted based on content
1) color
2) Texture
3) Shape
4) Position, etc.

4.13.1 Image Features and Distance Function

University Questions

Q. Explain the feature extraction and distance function for 2D color image. **SPPU : May 13, 8 Marks**

Q. Explain GEMINI approach for feature extraction and distance function for 2D color image. **SPPU : May 14, 8 Marks**

- Here we will focus on the color features, which can be resolves by the GEMINI approach. (1) convert to pixels (step)
- For color we compute a K-element color histogram for each item and scene where K = 256 or 64 colors. (2)
- Each component in the color histogram is the percentage of lixers that are most similar to that.
- Fig. 4.13.1 gives an example of such histogram of a fictitious photograph of a sensed, there are many red, pink, orange and purple pixels but only few white and green ones.
- Once these histograms are computed. One method to measure the distance between two histograms (K + 1 vectors) $\vec{x}$ and $\vec{y}$ is given by : But more features — more time, more complex

$$d^2_{hist}(\vec{x}, \vec{y}) = (\vec{x} - \vec{y})^t A (\vec{x} - \vec{y})$$

$$d^2_{hist}(\vec{x}, \vec{y}) = \sum_{i}^{K} \sum_{j}^{K} a_{ij} (x_i - y_i)(x_j - y_j)$$

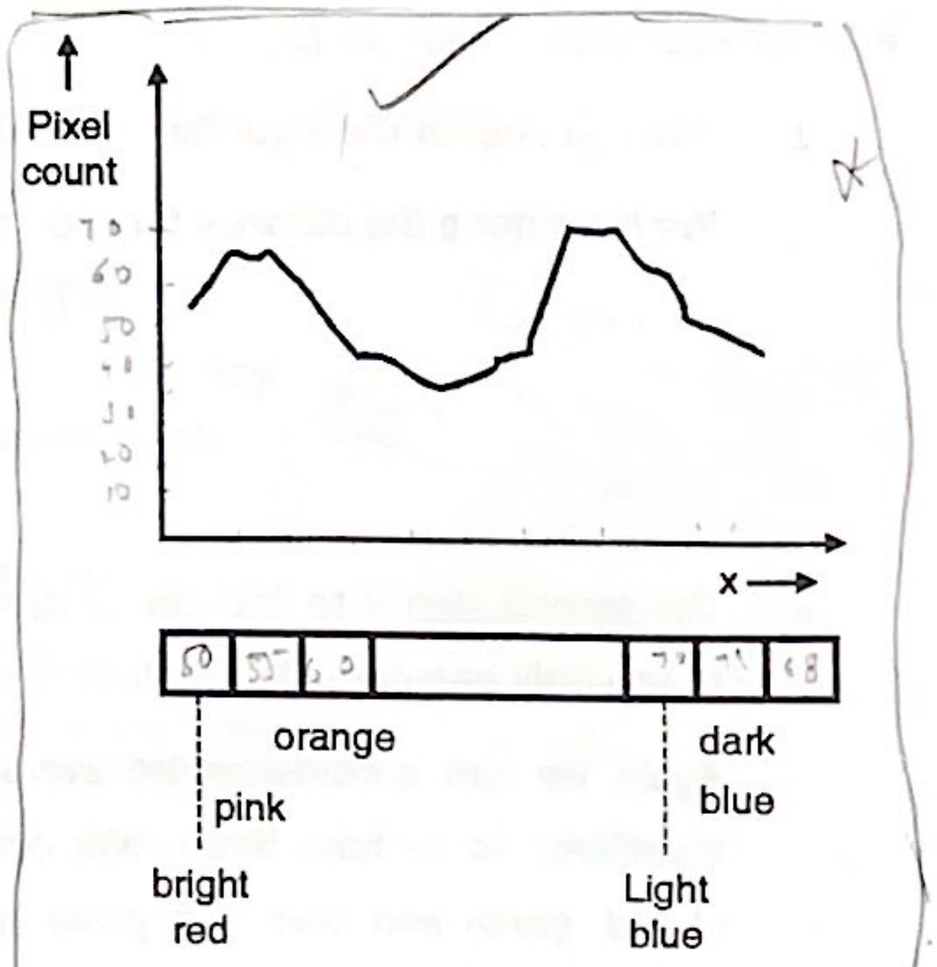


Fig. 4.13.1 : Example of a color histogram of a fictitious sensed photographs

- Where the superscript t indicates matrix transposition and the color-to-color similarity matrix A has entries a_{ij} which describe the similarity between color i and color j.

4.13.2 Lower Bounding

- For applying the GEMINI method for color indexing there are two obstacles :
  1. Dimensionality curse
  2. Quadratic nature of the distance function.
- The distance function in the feature space is a full quadratic form which involves all cross terms.
- Such distance function is expensive to compute than a Euclidean distance and it also precludes efficient implementation of commonly used spatial access method.
- Fig. 4.13.2 shows the illustration of the 'cross-talk' between two color histogram.

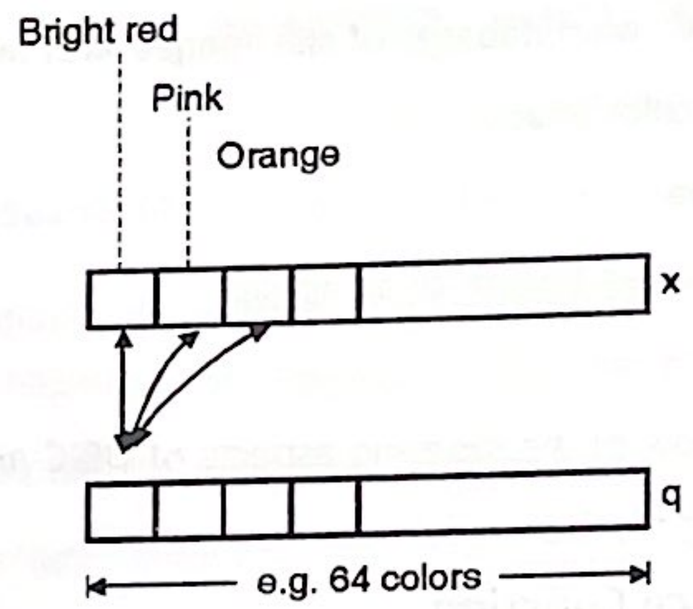


Fig. 4.13.2 : Cross-talk between two color histograms

- To compute the distance between two color histograms $\vec{x}$ and $\vec{q}$, the bright-red component of $\vec{x}$ has to be compared not only to the bright-red component of $\vec{q}$ but also to the pink and orange components of $\vec{q}$.
- To resolve the cross-talk problem, we will try to apply the GEMINI approach as follows.
  1. The first step of the algorithm is to calculate the distance function between two objects.

     We have done the distance function calculation between two color images and is given by the equation.

$$D(\,) = d_{hist}(\,) = (\vec{x} - \vec{y})^t A (\vec{x} - \vec{y})$$

$$= \sum_{i}^{K} \sum_{j}^{K} a_{ij} (x_i - y_i) (x_j - y_j)$$

  2. The second step is to find one or more numerical features whose Eudidean question is like : If we are allowed to use only one numerical feature to describe each color image. what should this feature be ?

     Again we can consider some average value or the first few coefficients of the two - dimensional DFT transform. As we have three color components i.e. Red, Green and Blue we will consider the average amount of red, green and blue in a given color image. So color of an individual pixel is described by the tripled (R, G, B) i.e. 'R'ed , 'G'reen and 'B'lue.

- The average color vector of an image or item $\vec{x} = (R_{avg}, G_{avg}, B_{avg})^t$ is defined as :

$$R_{avg} = (1/P) \sum_{P=1}^{P} R(p)$$

$$G_{avg} = (1/P) \sum_{P=1}^{P} G(p)$$

$$B_{avg} = (1/P) \sum_{P=1}^{P} B(p)$$

Where,

P → The number of pixels in the item

R(p), G(p), B(p) → Red, Green and Blue components and in the range 0-255 of the p^{th} pixel.

p → Pixels from 1 to P.

- Given the average colors $\vec{x}$ and $\vec{y}$ of two items, we can define $d_{avg}( )$ as the Euclidean distance between the three - dimensional average color vectors.

$$d^{2}_{avg}(\vec{x}, \vec{y}) = (\vec{x} - \vec{y})^{t} (\vec{x} - \vec{y})$$

3. The third step of the GEMINI algorithm is to prove that the simplified distance $d_{avg}( )$ lower bounds the actual distance $d_{hist}( )$. And this D true because of the application Quadratic Distance Bounding Theorem.

- So given a color query retrieval is proceeded by first filtering the set of images based on their average (R, G, B) color and then doing a final more accurate matching using their full K-element histogram.

4.13.3 Experiments using Bounding Theorem

- In this section we will present experimental results with GEMINI on color using the bounding theorem.
- The experiment compares the relative performance between simple sequential evaluation of d_{hist} for all database vectors and GEMINI approach in terms of CPU time and disk accesses.
- The experiments performs the simulations on a database of N = 924 color image histograms each of K = 256 colors, of assorted natural images and report the total and CPU times required by the methods.
- Results are shown in Fig. 4.13.2.
- Fig. 4.13.3 presents the total response time as a function of the selectivity.

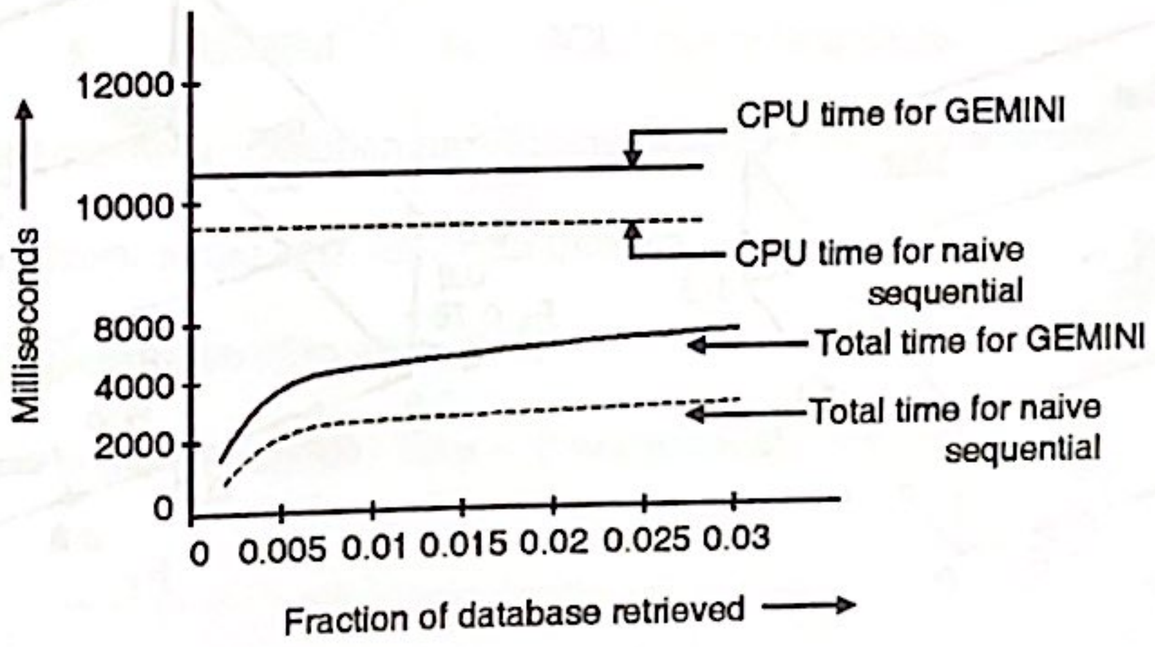


Fig. 4.13.3 : Response time VS selectivity for noise sequential and GEMINI

- Selectivity is the ratio of actual hits over the database size N.
- Fig. 4.13.3 also shows the CPU time for each method.

- So from the Fig. 4.13.3 it is clear that for a selectivity of 5% which would return ≈ 50 images to the user, the GEMINI method is much faster than the sequential computation of the histogram distances i.e. GEMINI method requires from a fraction of second up to ≈ 4 seconds while the naive sequential method requires consistently ≈ 10 seconds.
- For larger databases, the naive method will have a linearly increasing response time.
- Thus, finally following conclusions are drawn

1. GEMINI approach motivated a fast method using the average RGB distance.
2. GEMINI approach motivated a strong theorem i.e. QDB theorem to guarantee the correctness.
3. GEMINI approach resolves the cross-talk problem.
4. GEMINI approach resolves the dimensionality curse problem with no extra cost. If requires only f = 3 features as opposed to K = 64 or 256 that d_{hist} ( ) required.

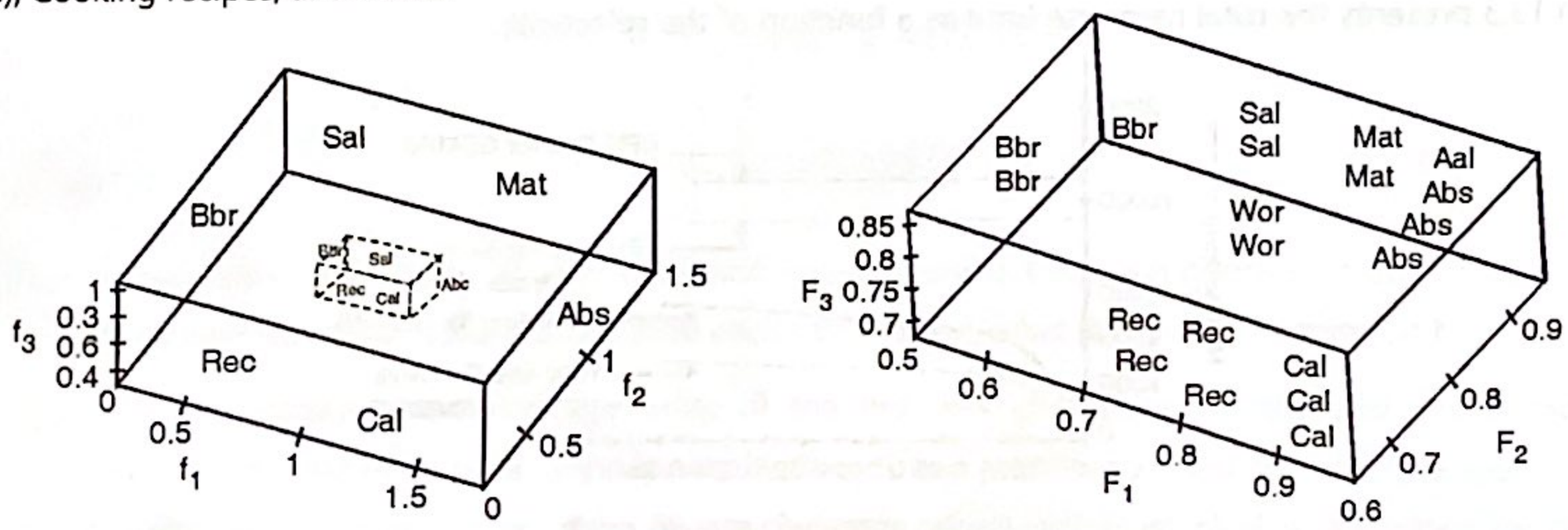
4.14 Automatic Feature Extraction

University Questions

Q. Write short note on : Automatic feature extraction. **SPPU : Dec.12, 8 Marks**

Q. Describe automatic feature extraction with the help of proper example. **SPPU : Dec. 13, 8 Marks**

- GEMINI approach is useful for any setting that we can extract features from.
- There are different algorithms for automatic feature extraction methods like the Multi-Dimensional Scaling (MDS) and Fast Map.
- Extracting features allows facilitating the use of off-the-shelf spatial access methods and visual data mining.
- He can plot a 2D or 3D projection of the data set and inspect it for correlations, clusters and other patterns.
- Fig. 4.14.1 shows the results of Fast Map on 35 documents of seven classes, after deriving K = 3 features or dimensions.
- In the Fig. 4.14.1, the classes included are basketball reports (Bbr), abstracts of computer science technical reports (Abs), Cooking recipes, and so on.



(a) Whole collection **(b) Magnification of the dashed box**

Fig. 4.14.1 : Collection of documents after Fast Map in 3-D space