Subspace tree-
Hierarchical linear subspace indexing method

- GEneric Multimedi INdexIng
  - DB in feature space
  - Range Query
- Linear subspace sequence method
  - DB in subspace
  - Generic constraints
  - Computing Cost
  - Orthogonal projection
  - Lower resolution images
  - Image Pyramid
  - Hierarchy of subspaces
- Ideas
Hierarchical Structure

- Our structure, a “tree” is organized in levels
  - Suppose each node belonging to a given level has the same number of children
  - The levels are labeled from 0 to L, level 0 is the root, level L are the leafs

\[ N(k) \text{ indicates for } k \in \{1, \ldots, L\} , \text{ the number of nodes which belong to level } k \]
  - Each node belonging to a given level has the same number of children

\[ N=N(L) \text{ is the number of leaves which represent the N ordered elements} \]
\[ N(0)=1 \text{ is the root itself} \]
Cost of a search in a tree

- Minimize the search time through such a tree

$$2 \cdot \text{cost}_{\text{average}} = \frac{N(L)}{N(L-1)} + \ldots + \frac{N(3)}{N(2)} + \frac{N(2)}{N(1)} + \frac{N(1)}{N(0)}$$

- Factor 2 appears because, on the average only half of the nodes at each level are encountered in a sequential search

For a tree:

$$2 \cdot \text{cost}_{\text{average}} = \frac{N(L)}{N(L-1)} + \ldots + \frac{N(3)}{N(2)} + \frac{N(2)}{N(1)} + \frac{N(1)}{N(0)}$$

$$L = \ln(N)$$

$$2 \cdot \text{cost}_{\text{min}} = e \cdot \ln(N)$$
For points in vector space

- Interested in designing a data structure, with the following objectives:
  - Space: $O(d \cdot n)$
  - Query time: $O(d \log(n))$
  - Data structure construction time is not important

- Is it possible?
- Metric indexes?
Images are characterized by feature vectors

- Color: color histogram
- Texture
- Shape: giving the shape of an object qualitative description which can be used to match other images
- Layout

Image matching

- Gray Images
  - 8-bit coding, 0 white, 255 black
- Color Images
  - color images 3-band RGB (interleaved modus)
- Two images $x$ and $y$ are similar, if
  \[ d(x, y) \leq \varepsilon \]
Scaling

- Bilinear method
  - interpolates linearly between the four closest lattice points

384*256 \rightarrow 240*180

DB

- 1000 images, scaled to size 240*180
  - http://wang.ist.psu.edu/docs/home.shtml

- \( DB[y] \approx 5.3 \text{ min} \)
  - Java/JAI (Java Advanced Imaging)
  - PowerMac G5, single processor 1.8Ghz, 2MB
Atmospheric similar images

- Images with the most similar color characteristic
Range query

\( \{x^{(i)} \in DB | i \in \{1..s\} \} \)

\( d[y]_n := \{d(x^{(i)}, y) | \forall n \in \{1..s\} : d[y]_n \leq d[y]_{n+1} \} \)

- Range query: search covers all points in the space whose Euclidean distance to the query \( y \) is smaller or equal to \( \varepsilon \)

\( DB[y]_\varepsilon := \{x^{(i)} \in DB | d[y]_n = d(x^{(i)}, y) \leq \varepsilon \} \)

\( \sigma = |DB[y]_\varepsilon| \)
Curse of dimensionality

- The metric indexes trees operate efficiently when the number of dimensions is small

- The growth of the number of dimensions has negative implications for the performance
  - failing with the dimensionality eventually reducing the search time to sequential scanning
  - problems arise from the fact that the volume of a sphere constant radius grows exponentially with increasing dimension

GEneric Multimedia INdexIng

- a feature extraction function maps the high dimensional objects into a low dimensional space
- objects that are very dissimilar in the feature space, are also very dissimilar in the original space

Christos Faloutsos
QBIC 1994
Lower bounding lemma

- \( d_{\text{feature}}(F(O_1), F(O_2)) \leq d(O_1, O_2) \)

- if distance of similar “objects“ is smaller or equal to \( \varepsilon \) in original space
- then it is as well smaller or equal \( \varepsilon \) in the feature space

Feature extracting function

1. Define a distance function
2. Find a feature extraction function \( F() \) that satisfies the bounding lemma

Example:

- Discrete Fourier Transform (DFT) preserve Euclidian distances between signals (Parseval's theorem)
- \( F() = \text{DTF} \) which keeps the first coefficients of the transform
Color histogram

- For an efficient retrieval of images based on their 3-band RGB (Red, Green, Blue) color histogram an efficient approximation to the histogram color distance is required

- This is achieved by the average color of an image

\[
R_{avg} = \sum_{p=1}^{N} R(p) \quad G_{avg} = \sum_{p=1}^{N} G(p) \quad B_{avg} = \sum_{p=1}^{N} B(p)
\]

- \( N \) represents the pixels in the image, \( R(p), G(p), B(p) \) are the red, green and blue components
The distance between two average color images is computed using the Euclidean distance function.

By the “Quadratic Distance Bounding” theorem it is guaranteed that the distance between vectors representing histograms is bigger or equal as the distance between histograms of average color images.

---

**DB in feature space**

\[ \{ F(\bar{x})^{(i)} \in F(DB) | i \in \{1..s}\} \]

\[ d[F(y)]_n := \{ d_{feature}(F(x^{(i)}), F(y)) | \forall n \in \{1..s\} : d[F(y)]_n \leq d[F(y)]_{n+1} \} \]

\[ F(DB|y)_n := \{ F(x)^{(i)} \in F(DB) | d[F(y)]_n = d_{feature}(F(x)^{(i)}, F(y)) \leq \varepsilon \} \]

\[ F(\sigma) = |F(DB|y)_n| \]
Range query

- points whose distance to the query point is smaller or equal to \( \varepsilon \) in the feature space are searched

- \( s \cdot f \) (\( f \) dimension of the feature space)

- false hits are filtered from the set of selected objects by comparison in the original space

- \( s \cdot m \geq F(\sigma) \cdot m + s \cdot f \) (\( m \) dimension of the original space)

- \( s \cdot (1 - \frac{f}{m}) \geq F(\sigma) \)

Linear subspace sequence method

- In this method, \( V \) is an \( m \)-dimensional vector space and \( F() \) is a linear mapping that obeys the lower bound lemma from the vector space \( V \) into an \( f \)-dimensional subspace \( U \)
In this case, the lower bounding lemma is extended:

- Let $O_1$ and $O_2$ be two objects; $F()$, the mapping of objects into $f$ dimensional subspace $U$
- $F()$ should satisfy the following formula for all objects, where $d$ is a distance function in the space $V$ and $d_U$ in the subspace $U$

$$d_U(F(O_1), F(O_2)) \leq d(F(O_1), F(O_2)) \leq d(O_1, O_2)$$

**Linear subspace sequence**

- Sequence of subspaces with, $V=U_0$ and $U_0 \supseteq U_1 \supseteq U_2 \supseteq \ldots \supseteq U_n$
- $\dim(U_0) > \dim(U_1) > \dim(U_2) \ldots > \dim(U_n)$
- Lower bounding lemma,

$$d(U_n(x_1), U_n(x_2)) \leq \ldots \leq d(U_1(x_1), U_1(x_2)) \leq d(U_0(x_1), U_0(x_2))$$
- Example,

$$\mathbb{R}^m \supset \mathbb{R}^{m-1} \supset \mathbb{R}^{m-2} \supset \ldots \supset \mathbb{R}^1$$
**DB in subspace**

\[ \{U_k(x)_i \in U_k(DB) | i \in \{1..s\}\} \]

\[ d[U_k(y)]_n := \{d(U_k(x)_i, U_k(y)) | \forall n \in \{1..s\} : d[U_k(y)]_n \leq d[U_k(y)]_{n+1}\} \]

\[ U_k(DB[x]) := \{U_k(x)_i \in U_k(DB) | d[U_k(y)]_n = d(U_k(x)_i, U_k(y)) \leq \varepsilon\} \]

\[ U_k(\sigma) = |U_k(DB[y])| \]

\[ U_0(\sigma) < U_1(\sigma) < U_2(\sigma) < \ldots < U_{(m)}(\sigma) < s \]

---

**Range query**

- points whose distance to the query point is smaller or equal to \( \varepsilon \) in the feature space are searched
  - \( s \cdot \text{dim}(U_i) \)
- false hits are filtered from the set of selected objects by comparison in the original space

\[ s \cdot m \geq U_1(\sigma) \cdot m + s \cdot \text{dim}(U_1) \]

\[ s \cdot (1 - \frac{\text{dim}(U_1)}{m}) \geq U_1(\sigma) \]

- apply it also for \( U_{n} \)

\[ s \cdot \text{dim}(U_1) \geq U_2(\sigma) \cdot \text{dim}(U_1) + s \cdot \text{dim}(U_2) \]

\[ s \cdot (1 - \frac{\text{dim}(U_2)}{\text{dim}(U_1)}) \geq U_2(\sigma) \]

- together:

\[ s \cdot m \geq U_1(\sigma) \cdot m + s \cdot \text{dim}(U_1) \geq U_1(\sigma) \cdot m + U_2(\sigma) \cdot \text{dim}(U_1) + s \cdot \text{dim}(U_2) \]
Constraints

- Savings compared when using only the subspace $U_2$ and the resulting $U_2(\sigma)$

$$U_2(\sigma) \cdot m + s \cdot \dim(U_2) \geq U_1(\sigma) \cdot m + U_2(\sigma) \cdot \dim(U_1) + s \cdot \dim(U_2)$$

$$U_2(\sigma) \cdot (1 - \frac{\dim(U_1)}{m}) \geq U_1(\sigma)$$

Generic constraints

for $k < n$

$$U_{(k+1)}(\sigma) \cdot (1 - \frac{\dim(U_k)}{\dim(U_{(k-1)})}) \geq U_k(\sigma)$$

for $k = n$

$$s \cdot (1 - \frac{\dim(U_n)}{\dim(U_{(n-1)})}) \geq U_n(\sigma)$$
Computing costs

\[ U_1(\sigma) \cdot m + U_2(\sigma) \cdot \text{dim}(U_1) + \ldots + s \cdot \text{dim}(U_n) = \]

\[ U_1(\sigma) \cdot \text{dim}(U_0) + U_2(\sigma) \cdot \text{dim}(U_1) + \ldots + s \cdot \text{dim}(U_n) = \]

\[ = \sum_{i=1}^{n} U_i(\sigma) \cdot \text{dim}(U_{i-1}) + s \cdot \text{dim}(U_n) \]

Orthogonal projection

\( P : \mathbb{R}^m \to U \)

- \( w^{(1)}, w^{(2)}, \ldots, w^{(m)} \) Orthonormal basis of \( \mathbb{R}^m \)
- \( w^{(1)}, w^{(2)}, \ldots, w^{(f)} \) Orthonormal basis of \( U \)
  (Gram-Schmidt orthogonalization process)

\[ \bar{x} = \sum_{i=1}^{f} \langle \bar{x}, w^{(i)} \rangle \cdot w^{(i)} + \sum_{i=f+1}^{m} \langle \bar{x}, w^{(i)} \rangle \cdot w^{(i)} \]

\[ P(\bar{x}) = \sum_{i=1}^{f} \langle \bar{x}, w^{(i)} \rangle \cdot w^{(i)} \]

\[ O(\bar{x}) = \sum_{i=f+1}^{m} \langle \bar{x}, w^{(i)} \rangle \cdot w^{(i)} \]

\[ ||\bar{x}||^2 = ||P(\bar{x})||^2 + ||O(\bar{x})||^2 \quad \Rightarrow \quad ||\bar{x}|| \geq ||P(\bar{x})|| \]
Orthogonal projection

- Corresponds to the mean value of the projected points
- Distance $d$ between projected points in $\mathbb{R}^m$ corresponds to the distance $d_u$ in the orthogonal subspace $U$ multiplied by a constant $c$

$$c = \sqrt{\frac{m}{f}}$$

$$U = \{(x_1, x_2) \in \mathbb{R}^2 \mid x_1 = x_2\}$$
Orthogonal projection

- Corresponds to the mean value of the projected points
- Distance $d$ between projected points in $\mathbb{R}^m$ corresponds to the distance $d_u$ in the orthogonal subspace $U$ multiplied by a constant $c$

$$c = \sqrt{\frac{m}{f}}$$

Color histogram

The average color of an image $x = (R_{avg}, G_{avg}, B_{avg})^T$ corresponds to an orthogonal projection.

Because of that, the equation

$$d_{avg}(F_{avg}(x_1), F_{avg}(x_2)) \leq d(x_1, x_2)$$

is valid, even

$$d_{avg}(F_{avg}(x_1), F_{avg}(x_2)) \leq d(F_{avg}(x_1), F_{avg}(x_2)) \leq d(x_1, x_2)$$

is valid.
Lower resolution images

- A lower resolution of an image corresponds to an orthogonal projection in rectangular windows, which define sub-images of an image.
- The image is tiled with rectangular windows \( W \) in which the mean value is computed (averaging filter).
- The arithmetic mean value computation in a window corresponds to an orthogonal projection of these values onto a bisecting line.
- Because of this, the different resolutions of an image correspond to a sequence of subspaces that satisfy the lower bounding lemma.

Figure 5: (a) Image of an elephant, with the size 210 × 180. (b) Image of the elephant, resolution 40 × 30. (c) The image of the elephant resolution 8 × 6. (d) The image of the elephant, resolution 4 × 3.
$U_3 (4*3)$

$U_2 (8*6)$
$U_1 (40*30)$

$U_0 (240*180)$
15.7 less complex
(would be ≈ 20 sec)
- 1000 images, scaled to size 240*180
- (DB[y] ≈ 5.3 min)

Hierarchy of subspaces

$U_0 \supset U_1 \supset U_2 \supset U_3$

- The distance between objects $d=d_{U_0}$ in the space $U_0$ can be obtained from the distance $d_{U_k}$ between objects in the orthogonal subspace $U_k$ by multiplying the distance $d_{U_k}$ by a constant

$$c_k = \sqrt{\frac{\text{dim}(U_0)}{\text{dim}(U_k)}}$$
Euclidian distance for a query $y$ to the elements of DB

$$d_{U_k}(F_{0,k}(O_1), F_{0,k}(O_2)) \leq d_{U_0}(F_{0,k}(O_1), F_{0,k}(O_2))$$

$$d[U_k(y)]_n := \{d(U_k(x^{(i)}), U_k(y)) \mid \forall n \in \{1..s\} : d[U_k(y)]_n \leq d[U_k(y)]_{n+1}$$

$$U_k(DB[y])_e := \{U_k(x)^{(i)}_n \in U_k(DB) \mid d[U_k(y)]_n = d(U_k(x)^{(i)}, U_k(y)) \leq \epsilon$$

$$U_k(\sigma) = |U_k(DB[y])_e|$$

$$U_0(\sigma) < U_1(\sigma) < \ldots < U_n(\sigma) < s.$$
Figure 6: (a) Image of a butterfly, with the size 128 × 96. (b) Image of the butterfly, resolution 32 × 24. (c) The image of the butterfly resolution 8 × 6. (d) The image of the butterfly, resolution 4 × 3.
Database consists of **9.876 web-crawled color images**

Mean Euclidian distance for a query $y$ to the elements of DB

- Mean computing costs using the hierarchical subspace method
- Error bars indicate the standard deviation
- The x-axis indicates the number of the most similar images which are retrieved and the y-axis, the computing costs
What has this to do with a tree?

\[ U_i(\sigma) = \frac{1}{\dim(U_{i+1})} \]

\[
\text{constant} \cdot d(U_0, U_{i+1}) \approx \frac{1}{\dim(U_{i+1})}
\]

\[ d(U_0, U_i) := \text{mean}(U_0(DB)) - \text{mean}(U_i(DB)) \]

\[
\text{mean}(U_i(DB)) := \left\{ \sum_{i \neq j} \frac{2 \cdot d(U_i(x_j^i), U_i(x_j^j))}{(s-1)s} \right\}
\]

\[
\text{cost} = \sum_{i=1}^{n} U_i(\sigma) \cdot \dim(U_{i-1}) + s \cdot \dim(U_n)
\]

\[
\text{cost} = \frac{\dim(U_0)}{\dim(U_1)} + \frac{\dim(U_1)}{\dim(U_2)} + \ldots + \frac{\dim(U_{n-1})}{\dim(U_n)} + \frac{\dim(U_n)}{1/s}
\]

Déjà vu

\[
2 \cdot \text{cost}_{\text{average}} = \frac{N(L)}{N(L-1)} + \ldots + \frac{N(3)}{N(2)} + \frac{N(2)}{N(1)} + \frac{N(1)}{N(0)}
\]

\[ L = \ln(N) \]

\[
2 \cdot \text{cost}_{\text{min}} = e \cdot \ln(N)
\]

\[
\text{cost} = \frac{\dim(U_0)}{\dim(U_1)} + \frac{\dim(U_1)}{\dim(U_2)} + \ldots + \frac{\dim(U_{n-1})}{\dim(U_n)} + \frac{\dim(U_n)}{1/s}
\]

\[ L := n + 1 \]

\[
L = \ln(s \cdot \dim(U_0))
\]

\[
\text{cost}_{\text{min}} = e \cdot \ln(\dim(U_0) \cdot s)
\]
Subspace tree

Instead of dividing the objects by a measuring function like the distance function in space by trees, hierarchical subspace indexing method divides the distances between the subspaces.

![Diagram showing subspace tree](image)

<table>
<thead>
<tr>
<th>Level L</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>N(0)</td>
</tr>
<tr>
<td>1</td>
<td>N(1), dim(U2)*s</td>
</tr>
<tr>
<td>2</td>
<td>N(2), dim(U1)</td>
</tr>
<tr>
<td>3</td>
<td>N(3), dim(U0)</td>
</tr>
</tbody>
</table>

\[
\text{cost} = \sum_{i=1}^{n} U_i(\sigma) \cdot \text{dim}(U_{i-1}) + s \cdot \text{dim}(U_n).
\]

\[
\text{cost}_{\text{min}} = e \cdot \ln(\text{dim}(U_0) \cdot s)
\]
Mean computing costs using the hierarchical subspace method
- Error bars indicate the standard deviation
- The x-axis indicates the number of the most similar images which are retrieved and the y-axis, the computing cost

\[
\text{mean}(U_{k}(DB)) = \frac{1}{s-1} \sum_{i=1}^{s} d[ U_{k}(DB) ]^i
\]

\[
d(U_0, U_k) := \text{mean}(U_0(DB)) - \text{mean}(U_k(DB))
\]

\[
U_k(\bar{\sigma}) \approx \frac{d(U_0, U_{k+1})}{\tan \alpha} + U_0(\bar{\sigma})
\]

\[
U_0(\bar{\sigma}) = 0
\]

\[
\frac{d(U_0, U_{k+1})}{\tan \alpha_0} \approx \text{constant} \cdot \frac{1}{\dim(U_{k+1})}
\]
Mean retrieval time of 26 medical images out of 12000

<table>
<thead>
<tr>
<th>method</th>
<th>comparisons</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>list matching</td>
<td>786,432,000</td>
<td>130.1 sec</td>
</tr>
<tr>
<td>subspace tree</td>
<td>3,322,536</td>
<td>1.2 sec</td>
</tr>
</tbody>
</table>
Subspace tree

- Instead of dividing the objects by a measuring function like the distance function in space by trees, hierarchical subspace indexing method divides the distances between the subspaces.
- $s$ number of objects, $\dim(U_0)$ their dimension, then the subspace tree converges to

$$L = \ln(s \cdot \dim(U_0))$$

$$\text{cost}_\text{min} = e \cdot \ln(\dim(U_0) \cdot s)$$

Questions

- Which other mappings beside the orthogonal mapping?
  - Example: PCA!
- What is the best hierarchy?
  - Depends on $\varepsilon$ and the number of elements…
- Other applications…
- Generalization of the Idea?
Literature