CBIR with a Subspacee-tree: Principal Component Analysis versus Averaging

Andreas Wichert and André Filipe da Silva Veríssimo
Department of Informatics
INESC-ID / IST - Technical University of Lisboa
Portugal
{andreas.wichert, andré.verissimo}@ist.utl.pt

Abstract

The subspace-tree is an indexing method for large multi-media databases. The search in such a tree starts at the subspace with the lowest dimension. In this subspace, the set of all possible similar images is determined. In the next subspace, additional metric information corresponding to a higher dimension is used to reduce this set. We compare theoretically and empirically data dependent mappings into subspaces (principal component analysis) with data independent mapping (averaging). The empirical experiments are performed on an image collection of thirty thousand images.

1 Introduction

In Content-based image retrieval (CBIR) in image or some drawn user input serves as a query example, and all similar images should be retrieved as results. An image query is performed through the generation of a weighted combination of features, and through its direct comparison with the features stored in the database. A similarity metric (e.g. the Euclidean distance) is then used to find the nearest neighbors of the query example in the feature vector space. In traditional content-based image retrieval methods, features describing important properties of the images are used, such as color, texture and shape [Flickner et al., 1995], [Smeulders et al., 2000], [Quack et al., 2004], [Dunckley, 2003].

In our approach, we combine the color information and its spatial distribution through simple image matching in high-dimensional space. We scale the digital images to a fixed size and map them into a 3-band RGB (Red, Green, Blue) representation. With this transformation, we are able to represent the images as vectors and to compute the Euclidian distance between any pair.

So, the used features are the scaled RGB images themselves, representing the color autocorrelogram and layout information. Two images $\vec{x}$ and $\vec{y}$ are similar if their distance is smaller or equal to ϵ, $d(\vec{x}, \vec{y}) \leq \epsilon$. The result of a range query computed by this method is a set of images that have spatial color characteristics that are similar to the query image.

The dimension of the resulting feature vector is extremely high, so an efficient high dimensional indexing method is required. The recently introduced sub-space tree [Wichert, 2008b], [Wichert, 2008a], [Wichert, 2009], [Wichert et al., 2010] does not suffer from the curse of dimensionality.

During content-based image retrieval, the search starts in the subspace with the lowest resolution of the images. In this subspace, the set of all possible similar

images is determined. In the next subspace, additional metric information corresponding to a higher resolution is used to reduce this set. This procedure is repeated until the similar images can be determined.

In this paper we compare data dependent mappings into subspaces such as principal component analysis with data independent mappings such as orthogonal projection (averaging). The main contribution of the paper is the theoretical proof as well as empirical experiments which highlight the conclusion, that orthogonal projection (averaging) with a constant is the best possible mapping. The paper is organized as follows:

- We describe the subspace-tree.
- Mapping functions into a subspace are introduced: principal component analysis and orthogonal mapping on the bisecting line. It is proven why the orthogonal mapping with a constant is the best possible mapping function.
- We make empirical comparison of PCA with the orthogonal mapping (averaging).
- We perform experiments on an image collection of thirty thousand images. Each image is represented by a 196608 dimensional vector.

2 Subspace-tree

In content-based multimedia indexing features describe multimedia objects. They are mapped into points in a high-dimensional feature space, and the search is based on points that are close to a given query point in this space. To speed up the search in the high dimensional feature space, indexing trees were proposed. In metric tree indexes, such as the R-trees, the d dimensional data space is recursively split by $d-1$ dimensional hyper-planes until the number of data s in a partition is below a certain threshold. There are many more tree structures, like the SS-tree, or R*-trees, X-trees, TV-trees, which use different heuristics to optimize the performance [Böhm et al., 2001]. However, the metric indexes trees operate efficiently only when the number of dimensions is small. The growth in the number of dimensions has negative implications in the performance of multidimensional index trees. These negative effects are named as the "curse of dimensionality." In high-dimensional spaces, a partition is performed only in a few dimensions touching the boundary of the data space in most dimensions. Because of these problems tree indexing methods deteriorate with the dimensionality eventually reducing the search time to sequential scanning.

The recently introduced sub-space tree [Wichert, 2008b], [Wichert, 2008a], [Wichert, 2009], [Wichert et al., 2010] does not suffer from the curse of dimensionality. The subspace tree divides the distances between the subspaces. These distances correspond to the values represented by the difference of the mean distance of all the objects in one space and a corresponding mean distance of the objects in a subspace.

V is an m-dimensional vector space and $F()$ is a linear mapping from the vector space V into an f-dimensional subspace U, $V \supset U$. Objects that are very dissimilar in the subspace are expected to be very dissimilar in the original space. (see Figure 1). Generic multimedia indexing (GEMINI) [Faloutsos et al., 1994],[Faloutsos, 1999] approach tries to find a feature extraction function that maps the high dimensional objects into a low dimensional space.

The size of the collection in the feature space depends on ϵ and the proportion between both spaces may reach the size of the entire database if the feature space is not carefully chosen. The lower bounding postulate which guarantees that no objects will be missed in the subspace space is expressed mathematically as follows

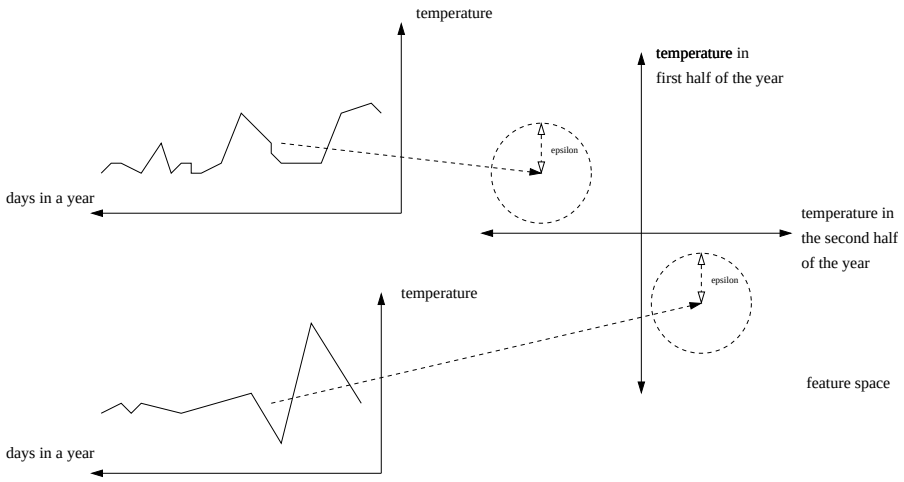

Figure 1: $F()$ maps the high dimensional objects from the space V into a low dimensional subspace U. The temperature in a city measured in days is mapped into the temperature of the first half of the year and the second half of the year. The distance between similar objects should be smaller or equal to ϵ. This tolerance is represented by a sphere with radius ϵ in the subspace.

Postulate 2.1 *Let O_1 and O_2 be two objects; $F()$, the mapping of objects into f dimensional subspace U should satisfy the following formula for all objects, where d is an Euclidian distance function $d = l_2$,in the space V and d_U is an Euclidian distance function in the subspace U:*

$$d_U(F(O_1), F(O_2)) \leq d(F(O_1), F(O_2)) \leq d(O_1, O_2). \tag{1}$$

There exists a constants c for which relation

$$d_U(F(O_1), F(O_2)) = c \cdot d(F(O_1), F(O_2)) \tag{2}$$

For the traditional GEMINI approach [Faloutsos et al., 1994] c is equal to one, and the distance function does not need to be Euclidean.

We can define a sequence of subspaces $U_0, U_1, U_2, \ldots, U_n$ with $V = U_0$ in which each subspace is a subspace of another space

$$U_0 \supset U_1 \supset U_2 \supset \ldots \supset U_n$$

and

$$dim(U_0) > dim(U_1) \ldots > dim(U_n).$$

Let DB be a database of s multimedia objects $\vec{x}^{(i)}$ represented by vectors of dimension m in which the index i is an explicit key identifying each object,

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

All s multimedia objects of DB are in space $V = U_0$, which is represented by $V(DB) = U_0(DB)$. The DB mapped by $F_{0,1}()$ from space U_0 to its subspace U_1 is indicated by $U_1(DB)$. A subspace U_k can be mapped from different spaces by different functions $\{F_{l,k}() | U_l \rightarrow U_k, l < k\}$ in contrast to the universal GEMINI approach [Faloutsos et al., 1994], in which the mapped DB only depends on the function $F()$. Because of this a notation is used which does not depends on the mapped function, but on the subspace U_k itself, $\{U_k(\vec{x})^{(i)} \in U_k(DB) | i \in \{1..s\}\}$;

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

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

with the size $U_k(\sigma) = |U_k(DB[y]_\epsilon)|$ and $U_0(\sigma) < U_1(\sigma) < \ldots < U_{(n)}(\sigma) < s$.

Corollary 2.1 *Let be $dim(U_0) := m$, the computing costs of the linear subspace sequence method are*

$$\sum_{i=1}^{n} U_i(\sigma) \cdot dim(U_{(i-1)}) + s \cdot dim(U_n). \tag{3}$$

The computing cost corresponds to the area defined by the sigma value and the preceding dimension $U_i(\sigma) \cdot dim(U_{i-1})$. To minimize the computing costs, the corresponding sum of areas has to be minimized using an adequate linear mapping $F()$ that meets all the required properties.

To estimate ϵ and its dependency on $U_k(\sigma)$, we define a mean sequence $d[U_k(DB)]_n$ which describes the characteristics of an image database of size s:

$$d_s[U_k(DB)]_n := \sum_{i=1}^{s} \frac{d[U_k(x^{(i)})]_n}{s}. \tag{4}$$

In the next section, we introduce mappings $F()$ that meet all the required properties.

2.1 Mapping function $F()$

We have to find a feature extraction function $F()$ that satisfies the bounding postulate 2.1. The function has to capture most of the characteristics of the objects in a low dimensional feature space. What would be a good mapping function $F()$?

Given Parseval's theorem, which states that the Discrete Fourier Transform (DFT) preserves Euclidian distances between signals, the DTF which keeps the first coefficients of the transform is an example of a feature function $F()$ [Faloutsos et al., 1994], [Faloutsos, 1999]. Accordingly, one can use any orthonormal transform like Wavelet transform because they all preserve the distance between the original and the transformed space. One can also use data dependent transforms as feature functions $F()$, such as the Karhunen Loève transform and PCA. However, they have to be recalculated as soon as new data arrives [Faloutsos et al., 1994].

PCA was developed using simple statistical tools such as standard deviation, covariance, eigenvector and eigenvalues. It uses these mathematical techniques to determine the correlated variables in a data set that change together in space. By knowing which variables are common in the data we can discard some variables without affecting the distance between objects and only keep the variables that make each object different and unique. Fundamentally, it analyses the data and extrapolates a new coordinate system, where the data is transformed. Then according to the variability of the data in each of the coordinates the PCA method will discard the coordinates that are not relevant and effectively reduce the dimension of the original data.

Usually the covariance matrix is determined that describes the variance of the data. Then the eigenvectors (principal components) and the eigenvalues of the covariance matrix are calculated. The eigenvectors define the Karhunen-Loève transform (KL transform). The Karhunen-Loève transform rotates the feature space into alignment with uncorrelated features. Each principal component (eigenvector) is associated with some variance represented by the eigenvalue. The idea of the

PCA is to retain only the significant principal components. Small eigenvalues correspond to less significant principal components. When using PCA for dimensionality reduction, it has to be decided how many eigenvectors to retain. The Kaiser criterion discards the components (eigenvectors), whose eigenvalues are smaller than one [Nadler and Smith, 1993, de Sá, 2001].

For evenly distributed data and an Euclidean distance function a good candidate for the mapping function $F()$is the orthogonal projection which corresponds to the computation of the mean value of the projected points (projection on the bisecting line).

Theorem 2.2 *(Lower bounding) Let O_1 and O_2 be two vectors; if $V = \mathbf{R^m}$ is a vector space and U is an f-dimensional subspace obtained by a projection and an Euclidian distance function $d = l_2$, then*

$$d_U(U(O_1), U(O_2)) \leq d(U(O_1), U(O_2)) \leq d(O_1, O_2). \tag{5}$$

Furthermore, we can map the computed metric distance d_U between objects in the f-dimensional orthogonal subspace U into the m-dimensional space V which contains the orthogonal subspace U by just multiplying the distance d_u by a constant $c = \sqrt{\frac{m}{f}}$,

$$d(U(O_1), U(O_2)) = \sqrt{\frac{m}{f}} \cdot d_U(U(O_1), U(O_2)). \tag{6}$$

Proof. An orthogonal projection P onto U is a mapping $P : \mathbf{R^m} \to U$. It orders every vector $\vec{x} \in \mathbf{R^m}$ a vector $P(\vec{x})$ with the shortest distance to $\vec{x} \in \mathbf{R^m}$. If $(w^{(1)}, w^{(2)}, \ldots, w^{(m)})$ is the orthonormalbasis of $\mathbf{R^m}$, and $(w^{(1)}, w^{(2)}, \ldots, w^{(f)})$ is the orthonormalbasis of U, then $\vec{x}$ can be represented by the unique decomposition

$$\vec{x} = \sum_{i=1}^{m} < \vec{x}, w^{(i)} > \cdot w^{(i)} = \sum_{i=1}^{f} < \vec{x}, w^{(i)} > \cdot w^{(i)} + \sum_{i=f+1}^{m} < \vec{x}, w^{(i)} > \cdot w^{(i)}$$

where U is a subset of $\mathbf{R^m}$. The orthogonal projection of $\vec{x}$ onto U can be represented by

$$P(\vec{x}) = \sum_{i=1}^{f} < \vec{x}, w^{(i)} > \cdot w^{(i)}$$

$$O(\vec{x})^{\perp} = \sum_{i=f+1}^{m} < \vec{x}, w^{(i)} > \cdot w^{(i)}$$

An orthogonal basis can be decomposed, for example, through the classical method named 'Gram-Schmidt orthogonalization' process. According to the Pythagorean theorem, $||\vec{x}||^2 = ||P(\vec{x})||^2 + ||O(\vec{x})^{\perp}||^2$; consequently, $||\vec{x}|| \geq ||P(\vec{x})||$, from which the lower bound lemma results [Lang, 1970], [Jedrzejek C., 1995].

For the second part of the theorem, we indicate that such an orthogonal projection exists. It is the mean value of the projected points, or the projection on the bisecting line. Suppose that we have a vector $\vec{a} = (a_1, a_2, ..a_k, .., a_m)$ with $a_1 = a_2 = ... = a_k = ... = a_m$ which represents the mean value of the projected points in m dimensional space. The length of this vector in Euclidian space of the dimension m is $\sqrt{m} \cdot a$. The length of $\vec{a} = a_1$ in one dimensional space is just a. The length of $\vec{a} = (a_1, a_2, .., a_k, .., a_f)$ with $a_1 = a_2 = ...a_k = ... = a_f$ in dimension f is $\sqrt{f} \cdot a$, so the division of the length between the two Euclidian spaces is $\sqrt{\frac{m}{f}}$.

If we compute several mean values (as done in the image pyramid), for example we compute a mean value of a m dimensional vector $\vec{y}$ in a window of size w. We

get a new reduced vector $\vec{z}$ of the dimension f with $f = \frac{m}{w}$. The length of the vector $\vec{z}$ is $l = ||z||_2$ and the length of the projected vector in m dimensional space is $\sqrt{w} \cdot l = \sqrt{\frac{m}{f}} \cdot l$.

Theorem 2.3 *(Best mapping) For evenly distributed data and an Euclidean distance function the best mapping function $F()$is the orthogonal projection which corresponds to the computation of the mean value of the projected points (projection on the bisecting line).*

The theorem is valid because the dimension of the distance in the subspace multiplied with the constant c is equivalent to the dimension of the original space.

Proof. To a given vector $\vec{x}$ of a dimension m we determine the closest vector $\vec{a} = (a_1, a_2, ..a_k, .., a_m)$ with $a_1 = a_2 = ... = a_k = ... = a_m = \alpha$ according to the Euclidian distance function. Each component is equal in vector $\vec{a}$. How do we choose the value of α? We want to minimize the distance $d(\vec{x}, \vec{a})$

$$\min_{\alpha} \left(\sqrt{(x_1 - \alpha)^2 + (x_2 - \alpha)^2 + ... + (x_m - \alpha)^2} \right)$$

$$0 = \frac{\partial d(\vec{x}, \vec{a})}{\partial \alpha} = \frac{m \cdot \alpha - (\sum_{i=1}^{m} x_i)}{\sqrt{m \cdot \alpha^2 + \sum_{i=1}^{m} x_i^2 - 2 \cdot \alpha \cdot (\sum_{i=1}^{m} x_i)}}$$

with the solution

$$\alpha = \frac{\sum_{i=1}^{m} x_i}{m}$$

which is the mean value of the vector $\vec{x}$. The value of the constant c corresponds to the dimension of the vectors, means $c = m$.

However, this assumption is only valid for evenly distributed data. For example, this assumption is not present in sparse representation, which in turn is present in the vector space model in information retrieval [Baeza-Yates and Ribeiro-Neto, 1999].

An example will be introduced, It is an intuitive demonstration of the relationship between the Pythagorean theorem and the orthogonal projection on the bisecting line. The orthogonal projection of points $\vec{x} = (x_1, x_2) \in \mathbf{R^2}$ on the bisecting line $U = \{(x_1, x_2) \in \mathbf{R^2} | x_1 = x_2\} = \{(x_1, x_1) = \mathbf{R^1}\}$ corresponds to the mean value of the projected points. The orthonormal basis of U is $x^{(1)} = (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$. The point $\vec{a} = (2, 4)$ is mapped into $P(\vec{a}) = 3$, and $\vec{b} = (7, 5)$ into $P(\vec{b}) = 6$. The distance in U is $d_u(P(\vec{a}), P(\vec{b})) = \sqrt{|6 - 3|^2}$, $c = \sqrt{2}$, so the distance in $\mathbf{R^2}$ is $d(P(\vec{a}), P(\vec{b})) = 3 \cdot \sqrt{2} \leq d(\vec{a}, \vec{b}) = \sqrt{26}$ (see Figure 2). It should be now clear, why the distance in the subspace multiplied with the constant c is equivalent to the dimension of the original space.

The subspace tree can be applied to multi-resolution techniques based on subsampling like the Fourier transform or Wavelet transform, given the Parseval's theorem that states that any orthonormal transform preserves the Euclidian distances between the original and the transformed space. The technique is not restricted to pixel-wise comparison between images. It can be applied for any data. The extracted features represented as a vector space and sub-sampled.

3 Empirical Experiments

We perform empirical experiments on an image collection of thirty thousand images based on subspace-tree. We compare two classes of feature functions $F()$, PCA and the orthogonal projection. The image collection was built using images downloaded

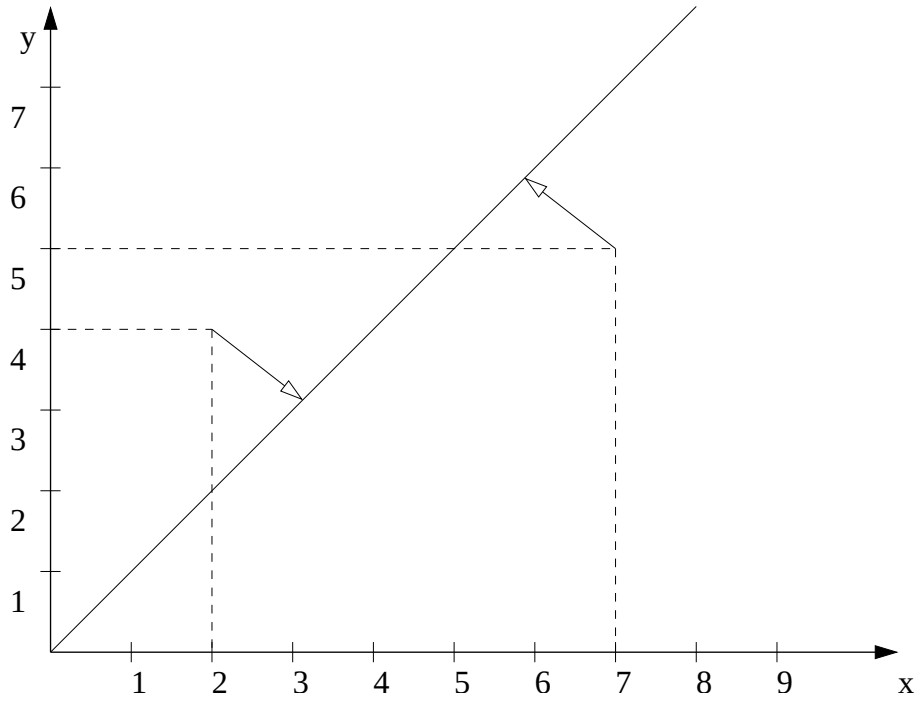

Figure 2: For example, the orthogonal projection of points $\vec{x} = (x_1, x_2) \in \mathbf{R^2}$ on the bisecting line $U = \{(x_1, x_2) \in \mathbf{R^2} | x_1 = x_2\} = \{(x_1, x_1) = \mathbf{R^1}\}$ corresponds to the mean value of the projected points. $\vec{a} = (2, 4)$ is mapped into $P(\vec{a}) = 3$, and $\vec{b} = (7, 5)$ into $P(\vec{b}) = 6$.

from flickr.com. All the downloaded images were published under a creative commons license that allows for non-commercial use that include academic and research purposes. Two images $\vec{x}$ and $\vec{y}$ are similar if their distance is smaller or equal to ϵ, $d(\vec{x}, \vec{y}) \leq \epsilon$. The result of a range query computed by this method is a set of images that have spatial color characteristics that are similar to the query image. We scale the images to the size 32×32 (in order to fit the memory requirements for PCA). We store 3-band RGB information for each pixel, that range from 0 to 255 resulting in 30000 vectors of size $32 \times 32 \times 3 = 3072$.

We compare PCA with multi-resolution technique on a example of the image pyramid. The representation of images at several resolutions corresponds to a structure which is called "image pyramid" in digital image processing [Burt and Adelson, 1983], [Gonzales and Woods, 2001]. The base of the pyramid contains an image with a high-resolution, its apex being the low-resolution approximation of the image.

3.1 Principal Components Analysis

The Kaiser criterion discards the components (eigenvectors), whose eigenvalues are smaller than one. This approach does not allow us to choose the resulting number of principal components as it is dependent on the variability of the original data. Applying the PCA with the Kaiser criterion to the whole set does not discard any principal components, suggesting that the data set is evenly distributed.

Because of this we use a heuristic function with different random samples of the collection as input for the PCA. Then choose how many principal components to keep. Table 1 shows the results and the samples that were chosen for the next step of the tests and the number of principal components. As can be seen, a sample of

9000 images does not lead to any reduction, a significant reduction is achieved with one tenth of the collection, 3000 images.

Sample size	Resulting Principal Components
9000	3072
3000	2527
2000	1969
1500	1487
1000	995
500	498
100	99

Table 1: Principal Components generated for each of the sample sizes.

The KL transform is computed. The eigenvectors define the KL transform. Each principal component (eigenvector) is associated with some variance represented by the eigenvalue. The original space is defined by the rotation of the input space by the KL transform and will be indicated by $U_0 = \mathbf{R}^{\mathbf{3072}}$. The first subspace is defined by the 2527 principal components (eigenvectors) ordered by the eigenvalue size. The second subspace is defined by the 1969 principal components out of the 2527. A a sequence of subspaces is the sequence of real vector subspaces

$$U_0 = \mathbf{R}^{\mathbf{3072}} \supset U_1 = \mathbf{R}^{\mathbf{2527}} \supset U_2 = \mathbf{R}^{\mathbf{1969}} \supset \ldots \supset U_6 = \mathbf{R}^{\mathbf{99}}$$

formed by the mapping from one subspace to another which always sets the discarded principal components to 0.

In Figure 3, we see that the ε-value threshold can be applied to the subspace with 995 dimensions and higher, whereas the characteristics for the lower subspaces are below the threshold.

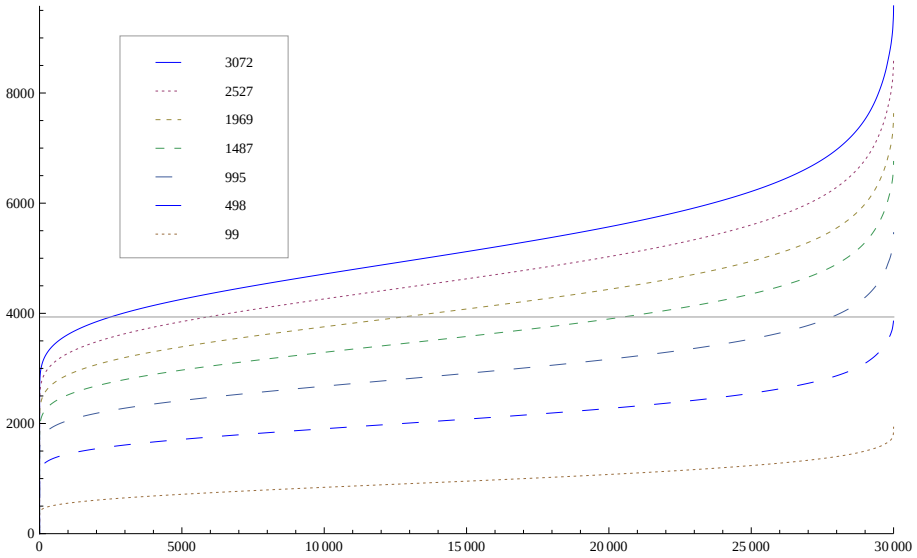

Figure 3: The x-axis indicates the most similar images which are retrieved and the y-axis the distance to the query image (Euclidian distance function). Characteristics plot using the PCA function and $\varepsilon = 3\,934.79$ which corresponds to 2500 retrieved images. For the definition of characteristic see the Equation 4.

The number of images retrieved on each subspace must be below a certain thresh-

old otherwise the use of the corresponding subspaces it will not give an advantage. This behaviour is present on many of the possible combinations and unless we use dimensions that are far apart we don't significantly reduce the number of operations necessary to process the query. Table 2 shows some of the possible ε-value that can be used and the number of images retrieved.

ε-value	Retrieved images
3225.95	200
3427.48	500
3616.26	1000
3745.73	1500
3848.21	2000
3934.79	2500

Table 2: Number of retrieved images for ε.

Consequently we can use the PCA method to reduce this collection to less than one third of the original dimension, and while the number of operations are reduced we still need to consider whether it is worth to use the smaller subspaces as few images are discarded.

For the ε-values < 3745.73 the best result is achieved when we start with two subspaces. The dimensions of the resulting subspaces are

$$dim(U_0) = 3072 > dim(U_1) = 2527 > dim(U_2) = 1969.$$

The average number of operations (which is 69.84 millions for $\varepsilon = 3225.954$) is better than when searching in the original space ($30000 * 3072 = 92.16$, 92.16 millions).

ε	list matching	one subspace	two subspaces	three subspaces	four subspaces	five subspaces
3225.95	92.16	78.37	**69.84**	72.86	89.32	103.61
3427.48	92.16	80.99	**77.95**	87.81	108.47	123.11
3616.26	92.16	**84.70**	88.02	104.31	127.88	142.71
3745.73	92.16	**88.02**	96.18	116.65	141.69	156.60
3848.21	92.16	**91.11**	103.24	126.78	152.76	167.70
3934.79	**92.16**	93.98	109.53	135.45	162.09	177.03

Table 3: Operations necessary for the query (in millions) in average for the PCA . The best choice of subspaces is indicated by bold.

3.2 PCA versus Mean Image Pyramid

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 of size $j \times j$ in which the mean value is computed (averaging filter). An example of an image pyramid is shown in the Figure 4.

The arithmetic mean value computation in a window corresponds to an orthogonal projection of these values onto a bisecting line. The representation of images in several resolutions corresponds to a structure which is called image pyramid in digital image processing [Burt and Adelson, 1983], [Gonzales and Woods, 2001]. The base of the pyramid contains an image with a high resolution (32×32), its apex being the low-resolution approximation of the image (4×4). The dimensions of the resulting subspaces are

$$dim(U_0) = 1024 \times 3 > dim(U_1) = 256 \times 3$$

Figure 4: Example of an image pyramid of an image corresponding to four different resolutions. First row represents images of different resolution with the same size, corresponding to the multiplication with the constant c. Second row represents images of different resolution which are not scaled. See as well Theorem 2.3.

$$> dim(U_2) = 64 \times 3 > dim(U_3) = 16 \times 3.$$

(we multiply by factor 3 for RGB color images.)

Let $\vec{x}^{(i)}$ and $\vec{x}^{(j)}$ be two objects from the image database, then for $k = 1, 2, 3$;

$$d_{U_k}(U_k(\vec{x}^{(i)}), U_k(\vec{x}^{(j)})) \leq d_{U_{k-1}}(U_k(\vec{x}^{(i)}), U_k(\vec{x}^{(j)})) \leq d_{U_{k-1}}(U_{k-1}(\vec{x}^{(i)}), U_{k-1}(\vec{x}^{(j)})) \tag{7}$$

Furthermore with $c_k = \sqrt{\frac{dim(U_{k-1})}{dim(U_k)}}$,

$$d_{U_{k-1}}(U_k(\vec{x}^{(i)}), U_k(\vec{x}^{(j)})) = c_k \cdot d_{U_k}(U_k(\vec{x}^{(i)}), U_k(\vec{x}^{(j)})) \tag{8}$$

with with $c_k = 2$, for $k = 1, 2, 3$ for the image databases

The hierarchical linear subspace method is able to achieve such a better performance because it applies for each subspace a constant that estimates the results in the original space, allowing the characteristics to be close together and converging to the same value, as shown in Figure 5. This allows the use of very small subspaces that greatly reduces the calculations, for example we can use the 4x4 subspace that has 48 dimensions, with a constant $c = 2^3 = 8$ that estimates the results in the original space. In Table 4 we compare the number of operations of PCA and orthogonal projection.

The results of the experiments indicate, that performing iteratively PCA on images is much less effective as comparing low resolution images.

Using PCA the constants c for the Equation $d_U(F(O_1), F(O_2)) = c \cdot d(F(O_1), F(O_2))$ is one.

3.3 CBIR based on Mean Image Pyramid

In the second phase we chose the 256×256 pixels resolution as the standard for which all 30000 images are scaled. The dimensions of the resulting subspaces are now

$$dim(U_0) = 65536 \times 3 > dim(U_1) = 16384 \times 3$$
$$> dim(U_2) = 4096 \times 3 > dim(U_3) = 3072 \times 3$$
$$> dim(U_3) = 256 \times 3 > dim(U_4) = 64 \times 3 > dim(U_5) = 16 \times 3$$

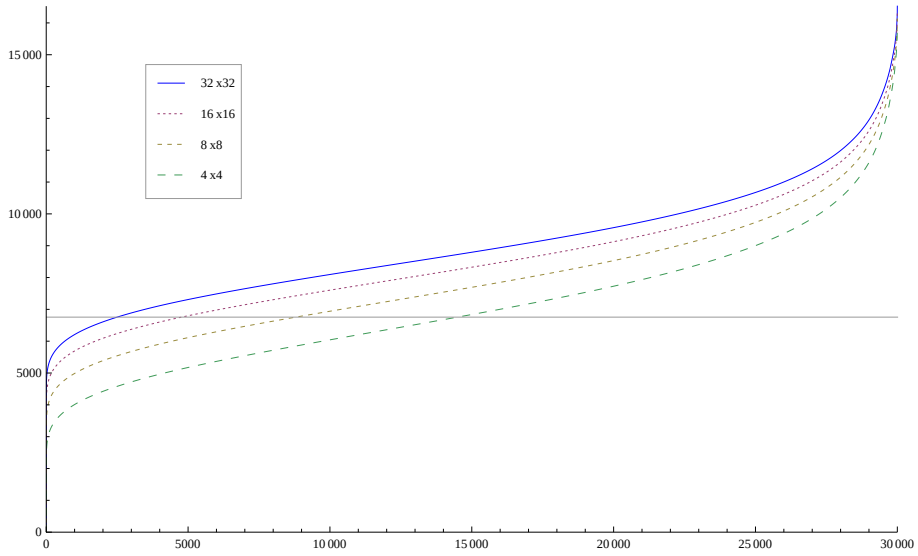

Figure 5: The x-axis indicates the most similar images which are retrieved and the y-axis the distance to the query image (Euclidian distance function). Characteristics plot using the orthogonal projection function and $\varepsilon = 3\,934.79$. For the definition of characteristic see the Equation 4.

ε-value	list matching	one subspace	two subspaces	**orthogonal projectionl**
3225.95	92.16	78.37	**69.84**	6.92
3427.48	92.16	80.99	**77.95**	10.51
3616.26	92.16	**84.70**	88.02	15.13
3745.73	92.16	**88.02**	96.18	18.99
3848.21	92.16	**91.11**	103.24	22.45
3934.79	**92.16**	93.98	109.53	25.64

Table 4: Number of operations necessary for the query (in millions) required in average for the PCA and the orthogonal projection. Orthogonal projection significantly less operations. The best choice of subspaces for PCA is indicated by bold, for orthogonal projection always three subspaces were used.

with with $c_k = 2$, for $k = 1, 2, 3, 4, 5$ for the image databases. To the 4x4 subspace that has 48 dimensions corresponds to a constant $c = 2^5 = 32$ that estimates the results in the original space. The characteristics are shown in the Figure 6. If we query the collection on the original space, list matching, the application needs to perform $256 \times 256 \times 3 \times 30\,000 = 5\,898\,240\,000$ pixel by pixel comparisons. Using the hierarchical linear subspace we can dramatically reduce this number by as much as 70.78 times, depending on the ε-value. By using this threshold, the results will on the average 200 images and require 3% of the comparisons of the original space, i.e. if the query was performed exclusively using this space. The hierarchical subspace method using the orthogonal projection can outperform the list matching method 70.78 times which is impressive as it can reduce queries from 3 minutes and 10 seconds to just 2.688 seconds on the average. Experiments were done on an Apple iMac running Mac OSX 10.4.11 operating system with 2.0GHz Core2 Duo processor and 1GB of RAM memory

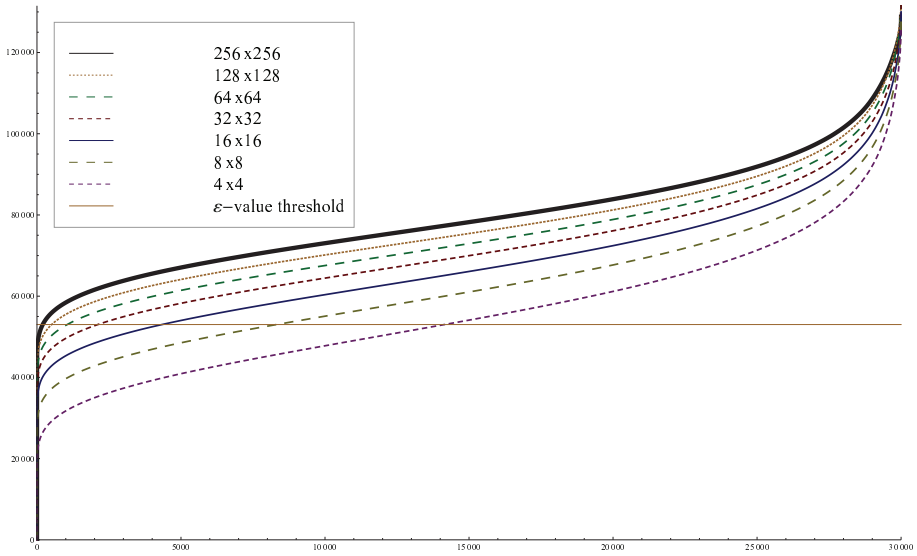

Figure 6: The x-axis indicates the most similar images which are retrieved and the y-axis the distance to the query image (Euclidian distance function). Characteristics plot with ε-value = 53 014.29. For the definition of characteristic see the Equation 4.

3.4 Related work

The best known content-based image retrieval system is the IBM QBIC (query by image content) search system [Niblack et al., 1993], [Flickner et al., 1995]. IBM QBIC uses features for color, texture and shape which are mapped into a feature vector. Similar features are used by the Oracle interMedia extender, which extends a relational database, so that it can perform CBIR [Dunckley, 2003]. The VORTEX system [Hove, 2004], on the other hand combines techniques from computer vision, with a thesaurus for objects and shapes description. In [Mirmehdi and Periasamy, 2001] human visual and perceptual systems are modeled. Perceptual color features and color texture features are extracted to describe the characteristics of perceptually derived regions in the image. Wavelet-based image indexing and searching WBIIS [Wang et al., 1997] is an image indexing and retrieval algorithm with partial sketch image searching capability for large image databases which is based on Wavelets. The algorithm characterizes the color variations over the spatial extent of the image in a manner that provides semantically meaningful image comparisons.

Known examples of CBIR systems which identify and annotate objects are [Blei and Jordan, 2003, Chen and Wang, 2004, Li and Wang, 2003b, Wang et al., 2001]. Jean et al. proposed [Jeon et al., 2003] an automatic approach to the annotation and retrieval of images based on a training set of images. It is assumed that regions in an image can be described using a small vocabulary of blobs. Blobs are generated from image features using clustering. In [Li and Wang, 2003a], categorized images are used to train a dictionary of hundreds of statistical models each representing a concept. Images of any given concept are regarded as instances of a stochastic process that characterizes the concept. To measure the extent of association between an image and the textual description of a concept, the likelihood of the occurrence of the image based on the characterizing stochastic process is computed.

Our approach is based on shallow rather than high level image understanding techniques. The goal of the system is not to identify objects in the image or to extract semantic information about it. Instead, images that have similar spatial

color characteristics to the query image, corresponding to the color autocorrelogram and layout information are determined. Our naive features are the scaled RGB images themselves, there is no need for weights (importance) between different features. However, the dimension of the resulting feature vector is extremely high, so an efficient high dimensional indexing method is required.

The extracted features of a CBIR system are mapped into points in a high-dimensional feature space, and the search is based on points that are close to a given query point in this space. For efficiency, these feature vectors are precomputed and stored. To speed up the search in the high dimensional feature space, indexing trees were proposed.

Traditional indexing trees can be described by two classes, trees derived from the kd-tree and the trees composed by derivatives of the R-tree. Trees in the first class divides the data space along predefined hyper-planes regardless of data distribution. The resulting regions are mutually disjoint and most of them do not represent any objects. In fact with the growing dimension of space we would require exponential many objects to fill the space. The second class tries to overcome this problem by dividing the data space according to the data distribution into overlapping regions, as described in the second section. An example of the second class is the M-tree [Paolo Ciaccia, 1997]. It performs exact retrieval with 10 dimensions. However its performance deteriorates in high dimensional spaces.

A solution to this problem consists of approximate queries which allow a relative error during retrieval. M-tree [Ciaccia and Patella, 2002] and A-tree [Sakurai et al., 2002] with approximate queries perform retrieval in dimensions of several hundreds. A-tree uses approximated MBR instead of a the MBR of the R-tree. Approximate metric trees like NV-trees [Olafsson et al., 2008] work with an acceptable error up to dimension 500.

The most successful approximate indexing method is based on hash tables. Locality sensitive hashing (LSH) [Andoni et al., 2006] works fast and stable with dimensions around 100. The method uses a family of locality-sensitive hash functions to hash nearby objects in the high-dimensional space into the same bucket. To perform a similarity search, the indexing method hashes a query object into a bucket, uses the data objects in the bucket as the candidate set of the results, and then ranks the candidate objects using the distance measure of the similarity search. There are several extensions of LSH, like for example Multi-Probe LSH [Lv et al., 2007] for example which reduces the space requirements for hash tables.

An alternative approach maps high dimensional data (dimension 10 to 270) into a space of dimension one. In one dimensional space, efficient tree techniques such as B-trees can be applied. Examples of such mappings are space filling curves [Zaniolo et al., 1997] and Pyramid Technique [Böhm et al., 2001]. Pyramid Technique divides the data space into two dimensional pyramids whose apexes lie at the center point. In a second step, each pyramid is cut into several slices parallel to the basis of the pyramid that form the data pages. The Pyramid Technique associates to each high dimensional point a single value, which is the distance from the point to the top of the pyramid, according to a specific dimension. The NB-tree [Fonseca and Jorge, 2003] maps the d dimensional space into a hyper-cube of the same dimension with edges of length one. In the next step the length of the corresponding points is determined, so that the objects can be ordered and represented by a simple B-tree.

3.5 Extremely high dimensional feature vector

It is important to note that testing the effectiveness of an image retrieval system can be subjective, since different users perceive the same image differently. Usually images are categorized, and although the categorization process may seem subjec-

tive, the efficiency of an indexing scheme may be determined objectively. The best feature extraction methods usually need large *feature vectors* to store a good representation of the image content. In our approach, we chose the largest possible *feature vector*, the image itself. We are comparing image pixel by pixel, thus being guaranteed to return those who are at a smaller distance from the query image. Usually one tries to reduce the size of feature vectors. This is because the large majority of indexing methods are very fast when dealing with low dimensional vectors but when that number increases, their performance deteriorates greatly (the *curse of dimensionality* [Böhm et al., 2001]). Fast multimedia queering leads to dilemma. Either the number of features has to be reduced and the quality of the results in unsatisfactory, or approximate queries are preformed leading to a relative error during retrieval.

A limitation is the dimension of the data, which is limited to the order of several hundreds. Because of this constraint neither of those techniques can be compared with a subspace tree, simply because they do not work in such a extreme high dimensional space (order of several thousands, 196608 dimension building an indexing structure for 30000 objects) performing *exact queries*. Because of the Parseval's theorem Wavelet transform on an image corresponds to a multiresolution-based decomposition on a nested sequence of linear spaces [Gonzales and Woods, 2001], and leads to similar results as the search on a mean value image pyramid.

4 Conclusion

Multiresolution theory is concerned with the representation and analysis of signals at more than one resolution. A powerful but conceptually simple structure for representation of images at more than one resolution is the image pyramid. In database applications its properties make it easy for users to access low quality versions of images during searches and later retrieve additional data to refine them. We have shown that neighborhood averaging which produces a mean pyramid is the best possible mapping into a subspace for evenly distributed data. This claim is a consequence of theorem 2.3 and its proof. It is as well supported by empirical experiments in section three. The claim is valid because the dimension of the distance in the subspace multiplied with the constant c is equivalent to the dimension of the original space. The principal components method performs poorly with a large sparse collection of images. The main problem with using the PCA as a mapping function is, that as the loss of information when reducing the dimension cannot be compensated by a constant (c is 1).

5 Acknowledgments

This work was supported by Fundao para a Cencia e Tecnologia (FCT) (INESC-ID multiannual funding) through the PIDDAC Program funds.

References

[Andoni et al., 2006] Andoni, A., Dater, M., Indyk, P., Immorlica, N., and Mirrokni, V. (2006). Locality-sensitive hashing using stable distributions. In MIT-Press, editor, *Nearest Neighbor Methods in Learning and Vision: Theory and Practice*, chapter 4. T. Darrell and P. Indyk and G. Shakhnarovich.

[Baeza-Yates and Ribeiro-Neto, 1999] Baeza-Yates, R. and Ribeiro-Neto, B. (1999). Modeling. In Baeza-Yates, R. and Ribeiro-Neto, B., editors, *Modern Information Retrieval*, chapter 2, pages 19–71. Addison-Wesley.

[Blei and Jordan, 2003] Blei, D. M. and Jordan, M. I. (2003). Modeling annotated data. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 127–134.

[Böhm et al., 2001] Böhm, C., Berchtold, S., and Kei, D., A. K. (2001). Searching in high-dimensional spaces—index structures for improving the performance of multimedia databases. *ACM Computing Surveys*, 33(3):322–373.

[Burt and Adelson, 1983] Burt, P. J. and Adelson, E. H. (1983). The laplacian pyramidas a compact image code. *IEEE Trans. Commin*, COM-31(4):532–540.

[Chen and Wang, 2004] Chen, Y. and Wang, J. Z. (2004). Image categorization by learning and reasoning with regions. *Journal of Machine Learning Research*, 5:913–939.

[Ciaccia and Patella, 2002] Ciaccia, P. and Patella, M. (2002). Searching in metric spaces with user-defined and approximate distances. *ACM Transactions on Database Systems*, 27(4).

[de Sá, 2001] de Sá, J. P. M. (2001). *Pattern Recognition: Concepts, Methods and Applications*. Springer-Verlag.

[Dunckley, 2003] Dunckley, L. (2003). *Multimedia Databases, An Object-Rational Approach*. Addison Wesley.

[Faloutsos, 1999] Faloutsos, C. (1999). Modern information retrieval. In Baeza-Yates, R. and Ribeiro-Neto, B., editors, *Modern Information Retrieval*, chapter 12, pages 345–365. Addison-Wesley.

[Faloutsos et al., 1994] Faloutsos, C., Barber, R., Flickner, M., Hafner, J., Niblack, W., Petkovic, D., and Equitz, W. (1994). Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3(3/4):231–262.

[Flickner et al., 1995] Flickner, M., Sawhney, H., Niblck, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., and Yanker, P. (1995). Query by image and video content the QBIC system. *IEEE Computer*, pages 23–32.

[Fonseca and Jorge, 2003] Fonseca, M. J. and Jorge, J. A. (2003). Indexing high-dimensional data for content-based retrieval in large databases. In *Proceedings of the 8th International Conference on Database Systems for Advanced Applications*, pages 267–274.

[Gonzales and Woods, 2001] Gonzales, R. C. and Woods, E. W. (2001). *Digital Image Processing*. Prentice Hall, second edition.

[Hove, 2004] Hove, L.-J. (2004). Extending image retrieval systems with a thesaurus for shapes. Master's thesis, Institute for Information and Media Sciences - University of Bergen.

[Jedrzejek C., 1995] Jedrzejek C., C. L. (1995). Fast closest codewordsearch algorithm for vector quantization. In *Proc. of the IEEE Information Theory Workshop ITW'95*.

[Jeon et al., 2003] Jeon, J., Lavrenko, V., and Manmatha, R. (2003). Automatic image annotation and retrieval using cross-media relevance models. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 119–126.

[Lang, 1970] Lang, S. (1970). *Linear Algebra*. Addison-Wesley.

[Li and Wang, 2003a] Li, J. and Wang, J. (2003a). Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Transactions on Pattern Analysis and Machine Learning*, 25(9):1075–1088.

[Li and Wang, 2003b] Li, J. and Wang, J. (2003b). Studying digital imagery of ancient paintings by mixtures of stochastic models. *IEEE Transactions on Pattern Analysis and Machine Learning*, pages 1–15.

[Lv et al., 2007] Lv, Q., Josephson, W., Wang, Z., Charikar, M., and Li, K. (2007). Multi-probe lsh: Efficient indexing for high-dimensional similarity search. In *Proceedings of the 33rd international conference on Very large data bases*, pages 950–961.

[Mirmehdi and Periasamy, 2001] Mirmehdi, M. and Periasamy, R. (2001). Cbir with perceptual region features. In *BMVC*.

[Nadler and Smith, 1993] Nadler, M. and Smith, Eric, P. (1993). *Pattern Recognition Engineering*. Wiley and Sons.

[Niblack et al., 1993] Niblack, W., Barber, R., Equitz, W., Flickner, M., Glasman, E. H., Petkovic, D., Yanker, P., Faloutsos, C., and Taubin, G. (1993). The qbic project: Querying images by content, using color,texture, and shape. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 173–187.

[Olafsson et al., 2008] Olafsson, A., Jonsson, B., and Amsaleg, L. (2008). Dynamic behavior of balanced nv-trees. In *International Workshop on Content-Based Multimedia Indexing Conference Proceedings, IEEE*, pages 174–183.

[Paolo Ciaccia, 1997] Paolo Ciaccia, Marco Patella, P. Z. (1997). M-tree: An efficient access method for similarity search in metric spaces. In *VLDB*, pages 426–435.

[Quack et al., 2004] Quack, T., Mönich, U., Thiele, L., and Manjunath, B. S. (2004). Cortina: a system for large-scale, content-based web image retrieval. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 508–511.

[Sakurai et al., 2002] Sakurai, Y., Yoshikawa, M., Uemura, S., and Kojima, H. (2002). Spatial indexing of high-dimensional data based on relative approximation. *VLDB Journal*, 11(2):93–108.

[Smeulders et al., 2000] Smeulders, A., Worring, M., Santini, S., Gupta, A., and Jain, R. (2000). Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380.

[Wang et al., 2001] Wang, J., Li, J., and Wiederhold, G. (2001). Simplicity: Semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(9):947–963.

[Wang et al., 1997] Wang, J. Z., Wiederhold, G., Firschein, O., and Wei, S. X. (1997). Content-based image indexing and searching using daubechies wavelets. *International Journal on Digital Libraries*.

[Wichert, 2008a] Wichert, A. (2008a). Content-based image retrieval by hierarchical linear subspace method. *Journal of Intelligent Information Systems*, 31(1):85–107.

[Wichert, 2008b] Wichert, A. (2008b). Subspace indexing for extremely high-dimensional cbir. In *International Workshop on Content-Based Multimedia Indexing Conference Proceedings, IEEE*, pages 330–338.

[Wichert, 2009] Wichert, A. (2009). Subspace tree. In *International Workshop on Content-Based Multimedia Indexing Conference Proceedings, IEEE*, pages 38–44.

[Wichert et al., 2010] Wichert, A., Teixeira, P., Santos, P., and Galhardas, H. (2010). Subspace tree: High dimensional multimedia indexing with logarithmic temporal complexity. *Journal of Intelligent Information Systems*, 35(3):495–516.

[Zaniolo et al., 1997] Zaniolo, C., Ceri, S., Snodgrass, R. T., Zicari, R., and Faloutsos, C. (1997). *Advanced Database Systems*. Morgan Kaufmann.