Web server for artists based on content-based queries that employ high-dimensional indexing of multimedia images

André Filipe da Silva Veríssimo

Abstract

Art portals are repositories for large multimedia collections that, in most cases, rely exclusively on annotations written by the artist itself or in collaboration between several users for image search. We plan to demonstrate an alternative method for querying a multimedia collection for similar image by using the image's content, called content-based image retrieval. To accomplish this goal we use relevant information about the image from raw pixel data without human input and interpretation.

The system implements a fully functional web portal offering artists the opportunity to display their work online to a broader audience. The portal will feature an image hosting service, keyword and content-based search options, a personal page for each user and all the necessary tools to manage the profile and respective artwork.

In this paper we focus on content-based image retrieval and high-dimensional indexing techniques such as the subspace tree indexing. We implemented a method that creates a pyramid of different dimensional spaces using a mapping function to generate them. This method speeds up the retrieval of similar objects by progressively discarding dissimilar objects as it visits the higher dimensional spaces.

Key words: Content-Based Image Retrieval, High-Dimensional Indexing, Art Portal, Hierarchical Linear Subspace, Orthogonal Projection, Principal Components Analysis

1. Introduction

There is an increasing number of portals on the web that host and present large collections of images, which range from stock photos and photography to digital art. These collections are very diverse and provide with search tools that mostly rely on information written by users. Image search engines have not kept up the pace with the collections they are searching and with user's necessities, focusing on the traditional keyword-based query that looks into text description when indexing, instead of the actual content of the image, objects, people, location, etc.

In this paper we present a fully functional art portal that presents the user with several methods of searching the collection, combining keyword and content-based search on image collection.

The search engine that will power the queries on the site will perform high-dimensional indexing of images based on their content provided by the artist. The data that is used to search by content is extracted automatically from the image without any user input. Although the goal is to read semantic content of the image, it is not possible using current technologies and existing research on a broad domain, therefore we chose to rely on low-level information to search the image collection by content.

2. High-dimension indexing schemes

In this section we will present different methods that reduce a high m-dimensional space while maintaining the distance between objects. It is important to use these methods when dealing with high-dimensional data (multimedia images) as comparing in the original space is a very expensive operation. There are many methods that can be applied

Email address: andre.verissimo@tagus.ist.utl.pt (André Filipe da Silva Veríssimo)

in order to index the high-dimensional objects, among them are the tree indexing techniques, for instance, the M-tree [2], A-tree [3], NB-tree [4]. Or other methods such as the space filling curves [5], Pyramid Technique [6] or the hierarchical linear subspace [7].

3. Hierarchical Linear Subspace method

The hierarchical linear subspace method [7] is based on the generic multimedia indexing (GEMINI) approach [8, 9] described before and makes use of the lower bounding lemma to create a pyramid of different dimensional spaces, or subspaces, with the original space in the bottom of the pyramid and the smallest on top. Each subspace is generated from the original space using a $F()$ mapping function that satisfies the lower bounding lemma [8, 9, 7] (see theorem 1 and reduces the original m-dimensional space V, to a f-dimensional subspace U. This allows for the application to query the collection and progressively reduce the number of images and calculations as each subspace is "visited".

A sequence of subspaces can be defined as $U_0, U_1, U_2, \ldots, U_n$ with $V = U_0$ in which each subspace is a subspace of another space, where $U_0 \supset U_1 \supset U_2 \supset \ldots \supset U_n$ and $dim(U_0) > dim(U_1) > dim(U_2) > \ldots > dim(U_n)$.

3.1. Orthogonal Projection

The orthogonal projection is a a linear mapping function $F()$ that calculate the mean value of the image's 3-band RGB to effectively reduce the original space V to the subspaces $U_1, U_2, U_3, \ldots, U_n$ and so on. This function is an orthonormal transformation and therefore satisfies the lower bounding lemma.

Theorem 3.1. *(Lower bounding lemma) Let O_1 and O_2 be two objects; if $V = \mathbf{R^m}$ is a vector space and U is a f-dimensional subspace obtained by a projection and an Euclidian distance function $d = l_2$, then*

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

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

$$d(U(O_1), U(O_2)) = \sqrt{\frac{m}{f}} \cdot d_U(U((O_1), U(O_2)). \tag{2}$$

This way not only can we reduce the original subspaces using the orthogonal projection but we can also approximate the distance between O_1 and O_2 to the original space.

(a) 256×256 space. **(b)** 40×40 subspace. **(c)** 8×8 subspace. **(d)** 4×4 subspace.

Figure 1: Image of a landscape, where 1(a) is the original space U_0 and 1(b) (U_1), 1(c) (U_2) and 1(d) (U_3) are the subspace after applying the orthogonal projection.

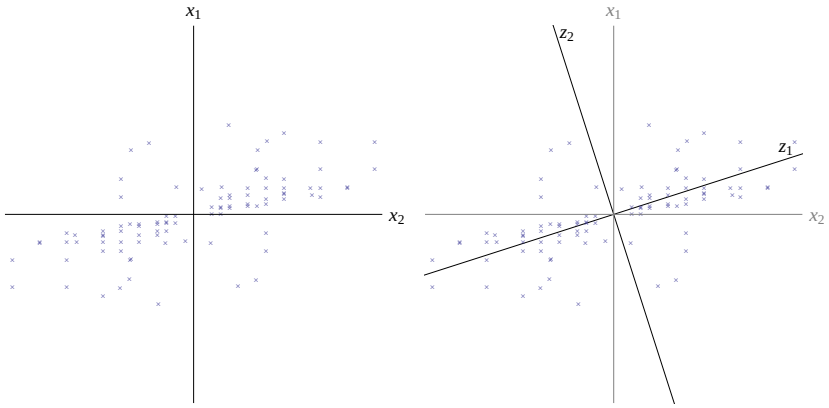
3.2. Principal Component Analysis

The principal component analysis method [10] (PCA) uses a data dependant transform to build its pyramid of low dimensional subspaces. It uses a $F()$ mapping function that is called the Karhunen-Loève transformation which

satisfies the lower bounding lemma [8]. This method has to be implemented under a condition, it needs to be recalculated either periodically or after new data is inserted. It depends on the collection's data to accurately reduce the dimensionality of multimedia objects.

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 [11]. 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 [12].

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.



(a) Original data. **(b)** New coordinate system.

Figure 2: The new coordinate system after applying the PCA method to a sample data.

4. Web Portal

We implemented a web-based application that supports the artist community in exhibiting their image portfolios and help them reach a broader audience. The application has all the basic features of popular art portals and also implement high-dimensional indexing techniques combined with CBIR queries.

We used a 3-tier client-server architecture that allowed us to separate the user interface from the application logic and database access layer. The CBIR methods and logic was developed as an weakly dependant module that can easily be adapted to other research projects and application.

We implemented a series of content-based search methods that range from a "brute force" approach (list matching) to high-dimensional indexing techniques that greatly reduce the computational costs when querying the collection and provide real-time results. Among the implemented methods are the (i) List matching, which performs a query on the original space of the objects; (ii) Colour Histograms, uses statistical information based on the colour distribution in multimedia images; (iii) Hierarchical Linear Subspace (see 3).

To keep with one of the major requirements of the applications, i.e. modular system, we decided to separate the image information in three entities, the metadata, the search data and the actual image. The original images as well as the generated subspaces were stored in disk while the metadata and other auxiliary data was kept in a MySQL relational database management system.

The application was developed using the Java foundation and WebObjects frameworks that allowed us to implement a modular system as described above. The actual application has three separate modules, the application's logic and interface, the utilities classes and the CBIR module that includes all the indexing functions and search queries.

5. Results

The experiments chapter presents the results of all tests done on the high-dimensional indexing techniques researched in this paper scope and on the web application that serves as the interface for the implemented algorithms. In line with the objectives that were defined in the beginning these experiments should test how the implemented algorithms perform, but also whether the application can be deployed for general use.

5.1. Test framework

The multimedia image collection was built using a set of 30.000 images was downloaded from flickr.com in order to simulate a real world scenario. All the downloaded images were published under a creative commons license that allows for non-commercial use that include academic and research purposes. In addition metadata information about the author and image was retrieved in order to comply with the terms of the license, when required.

The original resolution of the images could not be used for these experiments as they are very different and therefore a standard resolution was specified with all images being scaled to that size. We chose the 256×256 pixels resolution as the standard for which all images are scaled. While this process loses a lot of information, especially on the quality of the image's content, it still maintains all of the important features necessary for the implemented algorithms to work, such as colour distribution and objects' shapes. This resolution will store 3-band RGB information for each pixel, that range from 0 to 255 projecting the original colour spectrum to this range and colour mode.

We had to perform additional pre-processing for each image in the collection by applying the orthogonal projection mapping function, used in the hierarchical linear subspace method. The orthogonal projection used six different subspaces that represent the 128×128, 64×64, 32×32, 16×16, 8×8 and 4×4 resolutions using the mean value of the pixels' colour information.

For the principal components analysis tests we used a 32×32 scaled image in order to fit the memory requirements for PCA and available resources. In these tests we also used the 32×32, and smaller subspaces generated for using the orthogonal projection mapping in order to compare the two mapping functions.

5.2. Orthogonal Projection Results

The experiments that were done using this mapping function were performed with the main objective of documenting the hierarchical linear subspace method performance. But also how with just using a simple mapping function, that calculates the average pixel value, we can reduce the query time from minutes to just a few seconds proving that this function is mature and returns good results in a real world scenario that requires real-time results.

For this we chose to use the original space and six different subspaces with different dimensions in the experiments in order to test the mapping function and analyse the results. With the results we will be able to draw conclusions on whether it is worth to use this function. In the end we must balance the disk usage requirements with the accesses and comparisons that are done with the improvements in the results if there are any.

5.3. Metric indicators

For our experiments we chose to evaluate the performance of the method, the number of comparisons that are required as well as the number of images under the ε-value threshold. Both these indicators are objective and can clearly quantify the results, something that we cannot do for more subjective indicators such as the quality of the results.

Performance indicator. which indicates the overall time it takes to run a query of the collection.

Number of comparisons indicator. will estimate the number of operations being done in each query.

Number of images under the ε-value threshold. is directly related with the previous indicator as it shows how many images are discarded in each subspace.

Ratio indicator. will calculate how many times does one algorithm outperforms the other, i.e. if the hierarchical linear subspace method using the orthogonal projection mapping function requires 5 times less operations than the list matching, we can conclude that it is 5 times faster than the list matching method.

5.4. Characteristics

In order to estimate ε-value we defined a mean sequence (see 3), which describes the characteristics in the collection. It was impractical to calculate the characteristics using all images in the collection, as it would take approximately 70 days just for the original space, we then choose a 10% image sample that comprises of 3 000 images that present with a representative set of the collection. The images were chosen randomly from the collection making sure that group (defined by a keyword) had its share in the sample.

We used the Euclidean distance as the distance function for orthogonal projection mapping function experiments, these function was used to calculate the distance between the query image and every image in the collection.

The calculation of the characteristics was done by querying each image of the sample against the entire collection using list matching algorithm in the original space and respective subspaces. We then ordered each query's results from the smallest to the highest value and then calculated the data's mean value which describes the characteristics of an image collection in our system:

$$[U_k(DB)]_n := \sum_{i=1}^{s} \frac{d[U_k(x^{(i)})]_n}{s}. \tag{3}$$

The characteristics allows us to estimate ε-value as shown in figure 3 and determine in average how many images are returned by using any given ε-value.

We set the ε-value at 53 014.290 and as a result we can see that in average the hierarchical linear subspace method will discard 15 854 images in the 4×4 subspace, 5 828, 3 978 , 2 228, 1 061, 540 and 311 in the others, respectively. By using this threshold the results will in average return 200 images and require 3% of the comparisons of the original space, i.e. if the query was performed exclusively using this space. In fact, the application in average will only query the query image against 511 images in the original space. By using this method we can progressively discard irrelevant objects in the collection and minimise the comparisons that are made in the higher subspaces. The only drawback is that it will require additional storage space and disk accesses that adds complexity to the application and in the maintenance of the data integrity.

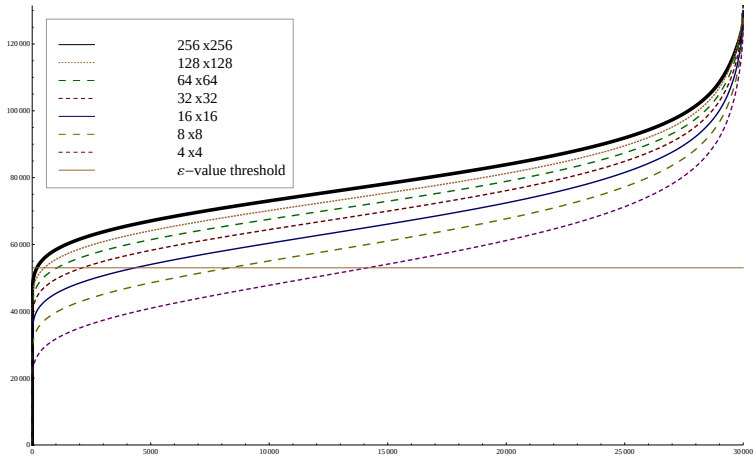


Figure 3: Characteristics plot with ε-value = 53 014.29.

This mapping function satisfies the lower bounding lemma [7] described in GEMINI [8, 9], therefore we can observe that the values of the any subspace are always lower than the other larger subspaces, including the values of the original space.

Table 1 shows how many images are discarded in each subspace and how it varies as the ε-value increases and more images fall into the threshold and are included in the results.

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.

ε-value	256x256	128x128	64x64	32x32	16x16	8x8	4x4
49 715.15	50	169	424	1 026	2 544	5 818	11 529
53 014.29	200	511	1 051	2 112	4 340	8 318	14 146
55 840.67	500	1 084	1 957	3 488	6 302	10 639	16 318
58 434.44	1 000	1 915	3 150	5 131	8 358	12 827	18 187
67 056.12	5 000	7 266	9 557	12 359	15 826	19 574	23 229
73 047.76	10 000	12 740	15 104	17 634	20 419	23 137	25 599

Table 1: Number of images below the ε-value threshold on each subspace.

ε-value	**Orthogonal Projection** (ms)	**Standard Deviation** (ms)	**Comparisons**	**Ratio**	**Predicted Ratio**
49 715.15	2 688	210	13 984 704	70.78	71.40
53 014.29	6 544	397	128 105 472	29.07	29.21
55 840.67	12 431	532	279 273 600	15.30	15.35
58 434.44	20 519	883	492 031 296	9.27	9.29
67 056.12	68 431	1 669	1 886 741 952	2.78	2.78
73 047.76	114 595	1 844	3 292 073 088	1.66	1.66

Table 2: Query time of both the hierarchical linear subspace method and list matching and respective ratio to determine how many times it outperforms the latter method.

When testing the different ε-value threshold we can observe that when using the lowest threshold this method 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. These results are verified from our tests using the overall time it takes for a query to return the results in table 2. In this table we can observe that the predicted ratio, which is calculated based on the savings in computational requirements is very similar to the real value.

The results and speed of this method depend heavily on the chosen threshold. If a high threshold is chosen more images will be included in the results and more comparisons will be required and more images will be queried in the higher dimensional subspaces. This reflects negatively on the query time but increase the number of results. On the other hand, if we lower the threshold the query will become faster as more images are filtered in the lower dimensional subspaces. However, the result set will have fewer images. It is required to weight the consequences of the different thresholds available in order to chose one that meets the goals of the application where this method is implemented.

In terms of performance the hierarchical linear subspace method presents encouraging results when combined with the orthogonal projection mapping function. However, an optimal threshold that balances the number of hits in the results with the query time is necessary. For situations where not all results need to be calculated this high-dimensional indexing technique and function can be deployed guaranteeing a good performance, especially with high ε-values.

5.5. PCA Results

The purpose of this experiment is to test whether PCA can become a viable mapping function for the hierarchical linear subspace method to reduce the high-dimensionality of multimedia image in CBIR systems. It will also be compared against the orthogonal projection function in order to determine which is the best mapping function for the hierarchical linear subspace. We will apply the PCA on a multimedia collection and then perform queries in order to test it.

The PCA behaves well with high dimensionality up a certain order, where the memory and computational requirements become massive. When calculating the principal components' matrix, the dimension size is crucial. If we have an image with MxN resolution with a 3-band RGB representation and 64 bits precision then the eigenvectors matrix's size is:

$$Dim = (M \times N \times 3)^2 \times 64 \tag{4}$$

		Memory usage (MB)	
Resolution	**Eigenvector Matrix cells**	64 bits precision	32 bits precision
256x256	38 654 705 664	294 912.00	147 456.00
128x128	2 415 919 104	18 432.00	9 216.00
64x64	150 994 944	1 152.00	576.00
32x32	9 437 184	72.00	36.00
16x16	589 824	4.05	2.25

Table 3: Eigenvector's matrix memory requirements for the different image resolutions.

Table 3 shows the memory usage for different image resolutions.

The PCA is most effective when dealing with similar and correlated data, as the more information the data has in common the more dimensions it can positively discard. Keeping only the variables that show the most variability. When dealing with sparse data the PCA method does not perform as well, since most variables are not correlated. Multimedia images are very different and unless a specific type of image and theme are targeted, such as portraits or landscapes, we may find that most of the variables cannot be discarded. Therefore when using very sparse data we may not be able to reduce the dimension of the collection to a desirable size that makes using the PCA useful and worth applying.

In order to avoid this problem with sparse collections we can try to find a smaller sample of images that is representative of the collection and apply the PCA based on that sample. We will be using the Euclidean distance to calculate the distance between the multimedia images.

Preparing the PCA data

We choose to use the co-variance method to calculate the principal components and then apply the Kaiser criteria to determine which to keep. 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. Because of this we needed to use different samples of the collection as input for the PCA and then choose which ones to keep for the tests. The 99, 498, 995, 1 487 and 2 527 subspaces were chosen as well as the original subspace (with 3 072 dimension).

Results of the experiments

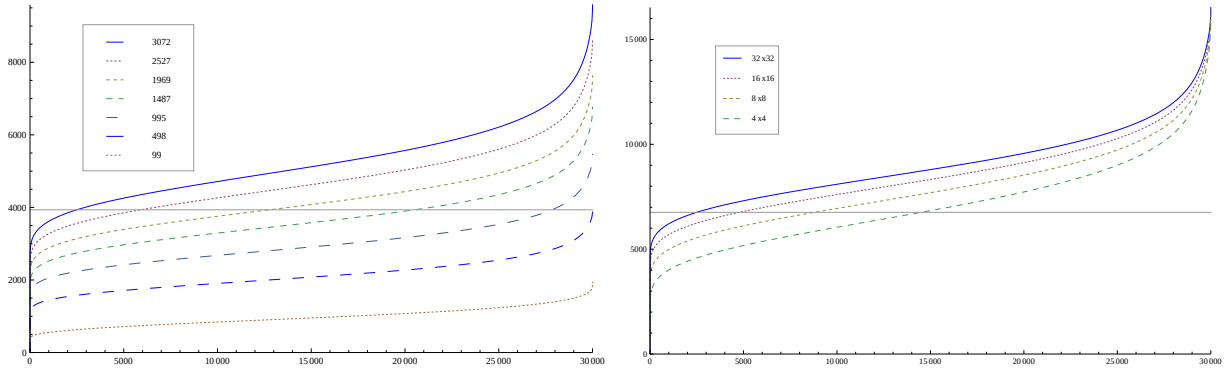
In order to compare the PCA against the hierarchical linear subspace method we use the characteristics data to analyse the results. This allowed us to predict the average number of retrieved images at each dimension that is determined by the ε-value, i.e. only the images with a Euclidean distance inferior to the ε-value are retrieved. Table 4 shows the time it took for to calculate the characteristics data, using a 10% sample from the collection, 3.000 of the 30.000 images in the collection.

PCA Sample	**Principal Components**	**Transformation to the New Coordinate System** (time)	**Applying the PCA** (time)
100	**99**	4 m 39.90 s	05 m 20.24 s
500	**498**	7 m 39.60 s	10 m 32.82 s
1000	**995**	9 m 30.97 s	15 m 43.70 s
1500	**1 487**	10 m 14.45 s	20 m 50.78 s
2000	**1 969**	11 m 23.84 s	26 m 4.36 s
3000	**2 527**	13 m 23.44 s	31 m 8.36 s
9000	**3 072**	14 m 56.26 s	36 m 24.3 s

Table 4: Performance of the PCA method under different samples.

In figure 4(b) 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. 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 significantly reduced we still need to consider whether it is worth to use the smaller subspaces as few images are discarded.



(a) using the PCA function.

(b) Using the orthogonal projection.

Figure 4: Characteristics plot for the respective functions with the ε-value was set at 3 934.79.

The number of images retrieved on each subspace must be below a certain threshold otherwise it is not worth to use that subspace in the query, as it will not give an advantage. The following formula calculates the number of images that must be retrieved in order for the dimension to be worth using.

$$number\,of\,images < size\,of\,collection \times \frac{(dim_i - dim_j)}{dim_i}, dim_i > dim_j \tag{5}$$

For example, the combination of using the 498 and 995 subspaces does not discard enough images to reduce the number of calculations. When the formula above is applied we see that the minimum number of images that must be discarded is 14.985, which in turn needs a very small ε-value that only returns two images in the original space and this is just the value where using this dimension is computational cheaper.

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 5 shows some of the possible ε-value that can be used and the number of images retrieved at each dimension.

	Images retrieved for each dimension in PCA						
ε-value	3072	2527	1969	1487	995	498	99
3 225.95	200	833	3 250	8 879	20 990	29 350	30 000
3 427.48	500	1 686	5 422	12 351	23 817	29 698	30 000
3 616.26	1 000	2 894	7 937	15 621	25 772	29 893	30 000
3 934.79	2 500	5 914	12 777	20 511	27 841	30 000	30 000

Table 5: Number of retrieved images with each dimension.

For the same ε-values the table below shows the average number of operations needed for the query, starting by comparing in that dimension and then using the result as the input for the next dimension. Among the different existing combinations the best result is achieved when we start with the 1 969 dimension and that only is valid until a certain ε-value. And although these results are slightly better than when searching in the original space, the number of required operations is still considerable higher when compared with the hierarchical linear subspace method.

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 4(b). 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.

ε-value	Operations necessary for the query (in millions)						
	PCA method						Orthogonal
	3 072	2 527	1 969	1 487	995	498	**Projection**
3 225.95	92.16	78.37	69.84	72.86	89.32	103.61	6.92
3 427.48	92.16	80.99	77.95	87.81	108.47	123.11	10.51
3 616.26	92.16	84.70	88.02	104.31	127.88	142.71	15.13
3 934.79	92.16	93.98	109.53	135.45	162.09	177.03	25.64

Table 6: Number of comparisons required in average for the PCA and the orthogonal projection.

6. Conclusion

The algorithm researched show great potential and encouraging results. From the experiments and existing research by other authors we concluded that the hierarchical linear subspace method is the most efficient and presented the best results from all algorithms. This was proven especially when using the orthogonal projection as a mapping function. The principal components analysis revealed too many limitations despite being able to reduce the dimension of the objects. While the list matching was too expensive and time consuming to be considered anything other than a good method to test the distance functions and to compare against.

The orthogonal projection mapping function delivered very promising results, but it still lacks the speed it is needed to be implemented on a collection that stores hundreds of thousands of images.

The principal components analysis failed completely as a mapping function as it is very limited, especially when it involves the higher dimensional spaces. We showed that the requirements of using covariance matrix method to calculate the principal components requires too much memory resources as the original dimensional space gets higher. In the experiments we could only test using a 3072 dimensional space and calculate the principal components, this is the equivalent of a image with a 32×32 resolution. These dimension does not have the necessary granularity to expect good results from queries. Notwithstanding these limitations, the real problem with this method is that the distance between objects in the transformed dimension is too different in comparison with the original space. And there is no method of approximate the results, as the orthogonal projection does by introducing the multiplication of a constant. As a result we can not eliminate sufficient objects in the smaller spaces to produce a saving on the number of comparisons that makes an impact on the query time.

Overall we can say with confidence that performing the hierarchical linear subspace method with the orthogonal projection mapping function is a viable method of querying a collection of high-dimensional objects. However as we said, the performance is directly linked to the ε-value threshold, which is extremely important as it need to be carefully chosen in order to limit the number of images in the result set.

References

[1] D. Forsyth, J. Malik, and R. Wilensky, "Searching for digital pictures," *Scientific American Magazine*, no. 276 (6), pp. 72–77, June 1997.

[2] P. Ciaccia and M. Patella, "Searching in metric spaces with user-defined and approximate distances," *ACM Transactions on Database Systems*, vol. 27, no. 4, 2002.

[3] Y. Sakurai, M. Yoshikawa, S. Uemura, and H. Kojima, "Spatial indexing of high-dimensional data based on relative approximation," *VLDB Journal*, vol. 11, no. 2, pp. 93–108, 2002.

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

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

[6] C. Böhm, S. Berchtold, and A. K. Kei, D., "Searching in high-dimensional spaces—index structures for improving the performance of multimedia databases," *ACM Computing Surveys*, vol. 33, no. 3, pp. 322–373, 2001.

[7] A. Wichert, "Content-based image retrieval by hierarchical linear subspace method," *Journal of Intelligent Information Systems*, vol. 31, no. 1, pp. 85–107, 2008. [Online]. Available: http://dx.doi.org/10.1007/s10844-007-0041-4

[8] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz, "Efficient and effective querying by image content," *Journal of Intelligent Information Systems*, vol. 3, no. 3/4, pp. 231–262, 1994. [Online]. Available: http://dx.doi.org/10.1007/BF00962238

[9] C. Faloutsos, "Modern information retrieval," in *Modern Information Retrieval*, R. Baeza-Yates and B. Ribeiro-Neto, Eds. Addison-Wesley, 1999, pp. 345–365.

[10] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine*, vol. 2, no. 6, pp. 559–572, 1901.

[11] I. Jolliffe, *Principal Component Analysis*, 2nd ed., ser. Springer Series in Statistics. Springer, Oct. 2002. [Online]. Available: http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/0387954422

[12] I. T. Jolliffe, "Discarding variables in a principal component analysis. i: Artificial data," *Applied Statistics*, p. 160–173, 1972.