Scalable Visual Descriptors for Content Based Image Retrieval

James Singleton

Department of Electrical and Electronic Engineering

The University of Sheffield, Sheffield, England

BEng, Electronic Engineering

Supervisor: Dr. Charith Abhayaratne

Second Marker: Dr. Luke Seed

May 2007

$\sim$ 6500 words

An electronic version of this paper (in full colour) is available on the accompanying disc.

Abstract

This project investigated the effects of scaling images with JPEG2000 emulation software on a CBIR system based on MPEG-7. A CBIR system was built that consisted of indexing and retrieval. The performance of the system was evaluated with standard metrics.

Various different versions of images were created with a process that simulates the behaviour of JPEG2000 with a wavelet transform. The resolution and compression quality of a selection of images were reduced and then MPEG-7 descriptors were extracted from them. The descriptors for various resolutions and qualities for the same image were then compared. Three MPEG-7 descriptors were used. They were the colour layout descriptor, scalable colour descriptor and edge histogram descriptor.

It was found that as the resolution and quality decreased, the descriptors became less similar to the originals; a trend which all descriptors were observed to follow. The colour layout descriptor was found to be the least sensitive to changes in resolution and quality.

It was found that not all of the data from an image is required in order to generate descriptors that can be used in image searches. The results of these searches, although not as good as ones done with the original images, provide satisfactory image matching.

Reducing the amount of data could be useful in environments where bandwidth or storage are scarce, such as is the case with mobile devices. Further work, expanding on these findings, has been recommended.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

I. List of Acronyms

ANMRR	Average Normalized Modified Retrieval Rank
AV	Audio Visual
CALIPH	Common And LIghtweight PHoto annotation
CBIR	Content Based Image Retrieval
CLD	Colour Layout Descriptor
CLI	Command Line Interface
CODEC	enCODer DECoder
CSD	Colour Structure Descriptor
CVS	Concurrent Versions System
DCD	Dominant Colour Descriptor
EHD	Edge Histogram Descriptor
EMIR	Experimental Meta data based Image Retrieval
GPL	General Public License
GUI	Graphical User Interface
HMMD	Hue Maximum Minimum Difference
HSV	Hue Saturation Value
IR	Information Retrieval
JPEG	Joint Photographic Experts Group
LIRE	Lucene Image REtrieval
MADIS	MPEG-7 Audio-visual Document Indexing System
MATLAB	MATrix LABoratory
MIRROR	MPEG-7 Image Retrieval Refinement based On Relevance feedback
MPEG-7	Motion Picture Experts Group standard 7
PC	Personal Computer
PNG	Portable Network Graphic
PR	Precision Recall
RGB	Red Green Blue
SCD	Scalable Colour Descriptor
WT	Wavelet Transform
XM	eXperimantal Model
XML	eXtensible Mark-up Language
YCbCr	Luminance and Chrominance

II. Introduction

A. Background

The Motion Picture Experts Group standard 7 (MPEG-7) defines a set of low-level visual descriptors for describing digital images and other media [1]. They are widely used in Content Based Image Retrieval (CBIR) systems. MPEG-7 is known as the 'Multimedia Content Description Interface'. It differs from the previous MPEG standards, which define how to encode Audio Visual (AV) data, as it describes the content of the AV material.

Scalable image compression methods, such as Joint Photographic Experts Group (JPEG) 2000, can be used to extract useful data from images without fully decoding them. Multiple versions of an image at a lower resolution or quality can be read from one file.

The objective of this project was to examine MPEG-7 colour descriptors, develop a system for matching similar images and then see if the matching performance of the descriptors was altered when using scalable image compression. This was done by testing the system on images with altered resolutions and qualities.

B. Motivation

The main reason for doing this project was to improve the efficiency of CBIR systems, by reducing the amount of information that is necessary to match similar images reliably. This can decrease the amount of computing power or bandwidth required, and hence increase the speed or reduce the needs of the system. If only part of the data from an image needs to be analyzed to achieve a reliable match then the remaining data doesn't need to be transmitted or processed.

C. Context

Although bandwidth, storage and processing power are plentiful in the desktop and server sectors these days, the same is not true in the portable world. When using a wireless link on a device with a low end processor, they are scarce resources. When in this restrained environment, such as when using a mobile camera phone, the amount of data needs to be limited. Clever compression schemes can be used to shrink the data, but this has an impact on the processing power required. A better solution would be to not require the complete set of data at all.

An example of an application could be taking a picture of a building on a mobile phone and wanting to find other images of it in an online database. The phone will have a low resolution camera and screen, limited storage, low processing power, and restricted bandwidth on the cellular network. The software on the phone may analyze part of the image data and send the results to a central server,

which would then send back low resolution versions of similar images. These would be suitable for viewing on the phone and would download quickly. User feedback could then be employed to refine the search. The full resolution, high quality images could be accessed later from a computer.

III. Specification

A modular system will be made consisting of many components. This will ensure that it is flexible and also make it easier to implement.

The data will be operated on in four phases; indexing, retrieval, performance evaluation and scaling. Indexing is the process whereby the descriptors are extracted from the image. This is done at the start, for the entire image database. Retrieval happens at the time of the query and compares the descriptors of many images. The query image will have descriptors extracted from it, but this will have been done at the start, as images from the database will be used for the queries. The performance evaluation metrics of the system will be calculated automatically. Lastly, scaled images will be used as the queries.

A. Indexing

Existing software will be used to extract descriptors from images and store them in a file. The file format will be eXtensible Mark-up Language (XML) to allow easy access to the data.

B. Retrieval

Software will be written to compare the descriptors of images in a database and find the closest match to a query image. It will be configureable with regards to how the various descriptors are weighted and combined. The images that have been matched will be ranked and the top matches will be displayed.

C. Performance Evaluation

More software will be written to analyse the ground truth data of the image database. This is information from a human perspective about how similar the different images look, such as information about the content of the images. The program will manipulate the data into a form that can be compared to the results of the matching component. Performance evaluation metrics, such as precision and recall, will then be calculated for the retrieval phase.

D. Scalability

The system that is built will be tested with scalable images. This will be done by creating many versions of images at different scales. The resolution and quality (level of compression) will be altered and the effects on retrieval recorded.

IV. Research

There are many descriptors defined in MPEG-7. This project mainly focuses on those concerned with colour. The following colour descriptors are part of the standard [2]:

- Colour Space Descriptor
- Scalable Colour Descriptor (SCD)
- Colour Layout Descriptor (CLD)
- Dominant Colour Descriptor (DCD)
- Colour Structure Descriptor (CSD)
- Group of Frames or Group of Pictures Descriptor

The Colour Space Descriptor is only used to define the colour space for another descriptor. There are six supported types, including Red Green Blue (RGB), Hue Saturation Value (HSV), Hue Maximum Minimum Difference (HMMD) and Luminance and Chrominance (YCbCr). The Group of Frames or Group of Pictures Descriptor is an extension to facilitate the description of video or an image set. The other four descriptors deal directly with image analysis.

Existing CBIR systems were investigated. One such system is MPEG-7 Image Retrieval Refinement based On Relevance feedback (MIRROR) [3] which has a web based interface. Another system which runs on a Windows Personal Computer (PC) is MUVIS [4]. There is also VisualsEEk [5], MPEG-7 Audio-visual Document Indexing System (MADIS) [6], PhotoMesa [7] and many others.

A. Existing Resources

Publicly available prototype software called the eXperimantal Model (XM) [8] was developed along with the standard. The source code is provided and it is written in the C++ programming language. The XM software was downloaded from http://www.lis.ei.tum.de/research/bv/topics/mmdb/e_mpeg7.html and thoroughly evaluated, but was found not to be suitable for use in this project.

The XM was compiled into a Command Line Interface (CLI) binary because a pre-compiled version could not be found. It was difficult to compile and the code was complex and hard to understand. Several other external libraries were required in order for it to run. These handle such things as image decoding and XML formatting.

The XM was tested on images, but the results obtained did not contain expected values. Table I shows the numbers generated when the XM was used to extract the DCD from an image (Image01.jpg in ./leaflesstrees/). The numbers were orders of magnitude out from what was expected as the values should have a maximum of 12 bits [1]. This was probably due to a bug in the software from an incomplete compile.

The latest version of the XM was checked out from the Concurrent Versions System (CVS) but this did not improve the results, as nothing had changed from the most recent (version 6.1) release with regard to colour descriptors. Lots of different settings were tried but without success. The lack of documentation accompanying the XM made it difficult to use.

10%	0	229763	0
8%	-8006272	0	0

TABLE I

THE DCD INDICES PRODUCED BY THE XM SOFTWARE

An alternative piece of software called Common And LIghtweight PHoto annotation (CALIPH) [9] [10] was found. It is open source, is licensed under the GNU General Public License (GPL) and is written in the Java programming language. CALIPH contains an retrieval module called Experimental Meta data based Image Retrieval (EMIR) but this was not used because the retrieval component was custom written. CALIPH can be downloaded from http://www.semanticmetadata.net and the latest version at the time, which was used in this project, was v0.9.23.

CALIPH was found to be much more suitable than the XM, as it reliably produced MPEG-7 data. It also saves the descriptors that it extracts from images into XML files. These are plain text files and are human readable. This maintained flexibility at later stages, as the data could easily be imported into lots of different applications for analysis. It was also important that the source code was available and could be modified if necessary. The code was examined to see how the descriptors were extracted. Documentation for CALIPH was extensive and helpful. CALIPH has a Graphical User Interface (GUI) that makes it easy to use, but this makes it harder to integrate it into a script than a CLI program. For this reason, the indexing is done off-line, i.e. separately from the rest of the system.

CALIPH can extract the following descriptors:

- SCD
- CLD
- DCD
- Edge Histogram Descriptor (EHD)

The DCD was not used, as when contacted, the author of the software said it was not fully implemented. Even so, table II shows that when the DCD was extracted from the same image as used for table I, the values that were generated were much more in line with what was expected [1].

The EHD is not a colour descriptor but it was included in the experiments to increase the scope. As the DCD was not able to be used and the EHD was supported by CALIPH it would have been wasteful not to use it. It was also simple to add support for it into the matching software as the format is similar to the other descriptors. It was useful to investigate if any discoveries made were also valid for a non-colour descriptor. The EHD can also improve the accuracy of matches, as texture is a prominent feature in natural images [1].

10%	12	21	27
8%	4	3	3

TABLE II

THE DCD INDICES PRODUCED BY CALIPH

Java programming was learned in order to modify the code. Ultimately it was not necessary to change anything about CALIPH but Java was used to implement the later components. Code written in it will run on lots of platforms (Windows, Mac and Linux) without modification. Java is also relevant to the project as it is prevalent on embedded devices such as mobile phones.

A development of CALIPH called Lucene Image REtrieval (LIRE) uses a database system and so accessing the data would be harder, hence it was not used. In a production system it may be better suited, as it might achieve a quicker retrieval than those that use XML as a storage medium.

An image database to use in the experiments was downloaded from the University of Washington [11] [12] (http://www.cs.washington.edu/research/imagedatabase/groundtruth/). The reasons for the choice of database were that it included a large selection of images, ground truth descriptions and it had been used in the project that was the precursor to this one [13].

There were other image databases found [14] [15] [4] but the chosen database was deemed sufficient. The low resolution images were not a set back as they were large enough to derive enough lower resolution versions. As an extra benefit, their small size allowed the images to be processed quickly, which sped up the experiments.

B. Matching Algorithm

The L_2Norm was used as a measure of how similar images were, and is defined in equation 2. This was chosen above the L_1Norm and other measures because due to the squaring, it only uses the magnitude of the values and ignores the sign. The L_1Norm could have been used if the sign of the values was ignored, see appendix A.

If X is a 1 dimensional array of values.

$$X = (x_1, x_2, x_3, ..., x_k)$$

Then the $L_N Norm$ is given by:

$$L_N(X) = \left(\sum_{i=1}^{k} x_i^N \right)^{\frac{1}{N}} \tag{1}$$

Where N takes any positive integer value:

$$N = 1 \rightarrow \infty$$

Then the $L_2 Norm$ is given by:

$$L_2(X) = \left(\sum_{i=1}^{k} x_i^2 \right)^{\frac{1}{2}} \tag{2}$$

i.e. all the values are squared, added together and the square root taken.

C. Performance Evaluation

It was important to be able to compare the results of the experiments with those done by others and have the data be meaningful. Therefore, standard performance evaluation metrics were used.

In the field of Information Retrieval (IR), common standards for determining the performance of a system are precision and recall [16]. Precision is defined as the number of relevant results retrieved over the total number of retrieved results. See equation 3.

$$precision = \frac{retrieved \cap relevant}{retrieved} \tag{3}$$

Recall is defined as the number of relevant results retrieved over the total number of relevant results. See equation 4.

$$recall = \frac{retrieved \cap relevant}{relevant} \tag{4}$$

Precision and recall should approximately be inversely proportional to each other. This is due to the nature of searching. If all relevant results are returned then it is very hard to not return some irrelevant ones. Conversely if only relevant results are returned then it is difficult to return all of them.

Precision and recall have limitations because a result has to be classified as either relevant or not. There is no partially relevant category. In other words they are binary and are not flexible or ‘fuzzy’. This means that thresholds have to be set in order to put the retrieved matches into these two groups. Where to put these thresholds is a hard decision because relevance is a subjective matter.

Another common performance metric is Average Normalized Modified Retrieval Rank (ANMRR) [17], but this was not used. Precision and recall were thought to be sufficient despite their shortcomings.

D. Scalability

A JPEG2000 emulator written in MATrix LABoratory (MATLAB) was provided by Dr. Abhayaratne. It incorporates a Wavelet Transform (WT) and is used to extract different sub bands from images. The image data is operated on in two phases; compression and extraction. Both components take a parameter of the number of wavelet decomposition levels. The extraction component also takes settings that determine the resolution and quality of the output image.

Five wavelet decomposition levels were chosen for the compression and extraction. A wrapper script, which made use of the MATLAB image processing toolbox, was written to automate the process of generating the scaled images, see appendix G.

V. Design Methodology and Experimental Procedure

The system was designed to be modular in nature, the purpose of which was to make it flexible. Each component is separate and can be modified or used on its own. MATLAB was used to tie the modules together and provide a graphical interface for displaying graphs and images. Due to the modularity, Microsoft Excel was used during experimentation to quickly compare results and plot graphs.

Figure 1 shows the system layout and figure 2 shows the key to the blocks. The shaded components were not made from scratch like the others, but acquired from other sources. The letters in square brackets relate to the methodology sections, e.g. C1 and C2 form the 'Display and Performance Evaluation' component.

A. Extraction

The Indexing stage consisted of the MPEG-7 data for the SCD, CLD and EHD being extracted by CALIPH and written into XML files. Once the data had been extracted, CALIPH was no longer used.

A feature in CALIPH called autopilot was used during the extraction process. It extracts the data from all the JPEG images in the currently selected image's directory. It does not work for Portable Network Graphic (PNG) files, but the files can simply be renamed to end in .jpg and CALIPH will process them correctly. If this 'hack' had not worked then CALIPH could have easily been modified to correct this bug.

B. Matching

The XML files are human readable and the data can be transfered to many programs for analysis. To make the process less arduous, a program was written in Java to automate it. See appendix B.

Fig. 1. The CBIR system layout

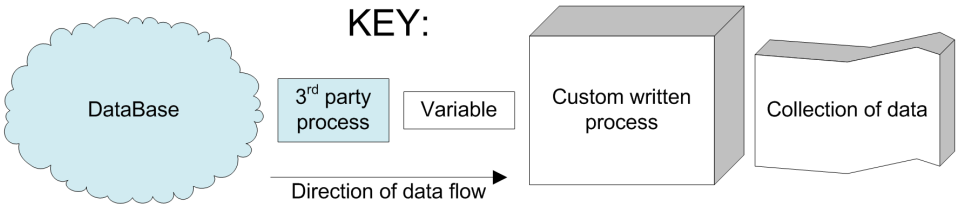


Fig. 2. The CBIR system layout key

The matching program has a CLI and takes an argument of the query file to compare. It can be called directly or from within MATLAB or a shell script if further automation is required. The program returns the L_2Norm distances between the image descriptors that it is instructed to process. The descriptor weighting and data set can be adjusted in the code.

C. Display and Performance Evaluation

The textual ground truth data that accompanied the image database consisted of a text file with a set of phrases for each image. See appendix F. Another CLI Java program was written to analyze the ground truth data. See appendix C. It assigns points for each image by comparing their phrases to those of a reference image. The more points an image has, the more phrases it has in common with the reference. Points are awarded for matching phrases and deducted for non-matching phrases.

A MATLAB script was written that calls the retrieval and ground truth evaluation programs, see appendix I. It then computes the precision and recall for various thresholds. It also ranks the returned images and displays them.

D. Scalability

The MATLAB script, mentioned above in the research, was used to generate different images from one source, see appendix G. It emulates the behaviour of a JPEG2000 image enCODer DECoder (CODEC). Nine versions of an image were generated: A full resolution, half resolution and quarter resolution image, each with three quality settings; loss-less (0), mildly compressed (5) and heavily compressed (8). The numbers refer to the settings in the MATLAB script that determine the level of compression that is used, i.e. the higher the number, the more data is thrown away. Example images are shown in appendix J.

Some images needed to be resized, to have dimensions that were factors of $2^5 = 32$ in order to work with the script at 5 wavelet decomposition levels. Different methods of reshaping the images to be compatible with the JPEG2000 emulator were investigated. Adding black bands to the bottom and

right of the image was tried as this was thought to not increase the energy of the image, although it did modify the descriptors.

The final solution was to use the 'imresize' command in the MATLAB image processing toolbox to resize the image, see appendix G. This altered the descriptors extracted from images least.

PNG files were used to store the images as it is a loss-less format and no errors would be introduced. Descriptors were then extracted from the re-scaled images using the indexing component. The descriptors were then analysed with the matching component.

VI. Retrieval Results

A large number of results were obtained and they cannot all be shown here. Two images were selected from the 'campusinfall' image set to demonstrate the findings.

To start with, each of the three descriptors were tested individually on the original images. In the figures showing the retrieved images the rankings go from left to right. The top five images are shown and the first is the query image. The tables show the L_2Norm values returned and the image number in the set that they relate to. The higher the value of the L_2Norm the less similar the image is to the query image.

A. CLD

The CLD differs from the other two descriptors as it has many components. The SCD and EHD are both stored as a single array on integers but the CLD uses six. These are for luminance and red and blue chrominance, each with one dc and many ac components. The dc component is stored separately because it is quantised differently to the ac components [1].

Various different weighting combinations were experimented on. In figure 3, figure 4, table III and table IV all six CLD components have an equal weighting.

The images returned for a query of 'Image06' when using only the CLD are shown in figure 3. The associated numerical data is shown in table III.

Fig. 3. The ranked retrieved images for Image06 with the CLD

The images returned for a query of 'Image04' when using only the CLD are shown in figure 4 and the associated numerical data is shown in table IV.

L2Norms	0	5.7442	6.0338	6.4353	6.5609
Image numbers	6	10	29	17	5

TABLE III

THE RANKINGS FOR IMAGE06 WITH THE CLD

Fig. 4. The ranked retrieved images for Image04 with the CLD

Upon visual inspection of the images returned by the algorithm, it was discovered that in general the images returned as the top results look similar to the query image. The ground truth also confirmed that the matching was operating correctly. See appendix F.

L2Norms	0	6.3102	6.4169	6.5014	6.5056
Image numbers	4	45	35	48	19

TABLE IV

THE RANKINGS FOR IMAGE04 WITH THE CLD

B. SCD

The images returned for a query of 'Image06' when using only the SCD are shown in figure 5. The associated numerical data is shown in table V.

Fig. 5. The ranked retrieved images for Image06 with the SCD

The images returned for a query of 'Image04' when using only the SCD are shown in figure 6 and the associated numerical data is shown in table VI.

L2Norms	0	22.4722	26.6833	28.688	33.1662
Image numbers	6	5	9	10	2

TABLE V

THE RANKINGS FOR IMAGE06 WITH THE SCD

Fig. 6. The ranked retrieved images for Image04 with the SCD

C. EHD

The images returned for a query of 'Image06' when using only the EHD are shown in figure 7. The associated numerical data is shown in table VII.

Fig. 7. The ranked retrieved images for Image06 with the EHD

The images returned for a query of 'Image04' when using only the EHD are shown in figure 8. The associated numerical data is shown in table VIII.

D. Combined Descriptors

The results of the first part were used, in conjunction with the ground truth data and visual inspection, to calibrate the system. This involved setting the weightings of the three descriptors

L2Norms	0	26.0768	28.3725	33.6601	33.7194
Image numbers	4	31	19	40	39

TABLE VI

THE RANKINGS FOR IMAGE04 WITH THE SCD

L2Norms	0	15.2971	15.3623	15.7797	16.2481
Image numbers	6	1	5	22	29

TABLE VII

THE RANKINGS FOR IMAGE06 WITH THE EHD

Fig. 8. The ranked retrieved images for Image04 with the EHD

used to the best combination for good matches.

Experiments were done to find the best combination of the descriptors. A weighting of 4 for the SCD, 2 for the CLD luminance, 1 for both CLD chrominance and 8 for the EHD was settled upon.

Figure 9, 10 and 11 show the individual descriptors with Precision Recall (PR) graphs and figure 12 shows the same for the final weighting. 'Image17' in the set 'campusinfall' is used as the query image in these results.

In the PR graphs, recall is on the x axis and precision is on the y axis. The four graphs show a decreasing ground truth threshold from left to right, i.e more images are made relevant. The images are ranked 1st, 2nd, 3rd and 4th from left to right respectively on the top line, and similarly from 5th through 8th on the bottom.

VII. SCALABILITY RESULTS

The images were re-scaled using the JPEG2000 emulator and had descriptors extracted from them. The results that follow use the descriptors from the scaled images to retrieve the original images from the database.

L2Norms	0	13.0384	13.2288	14.1774	14.3527
Image numbers	4	24	19	36	3

TABLE VIII

THE RANKINGS FOR IMAGE04 WITH THE EHD

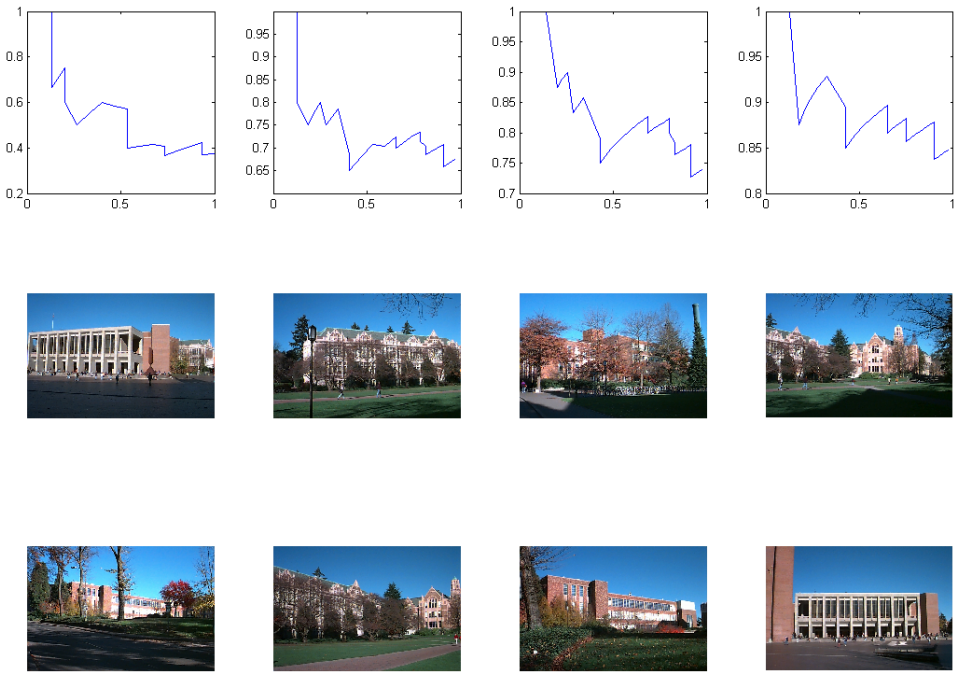


Fig. 9. The ranked retrieved images and PR graphs for the CLD

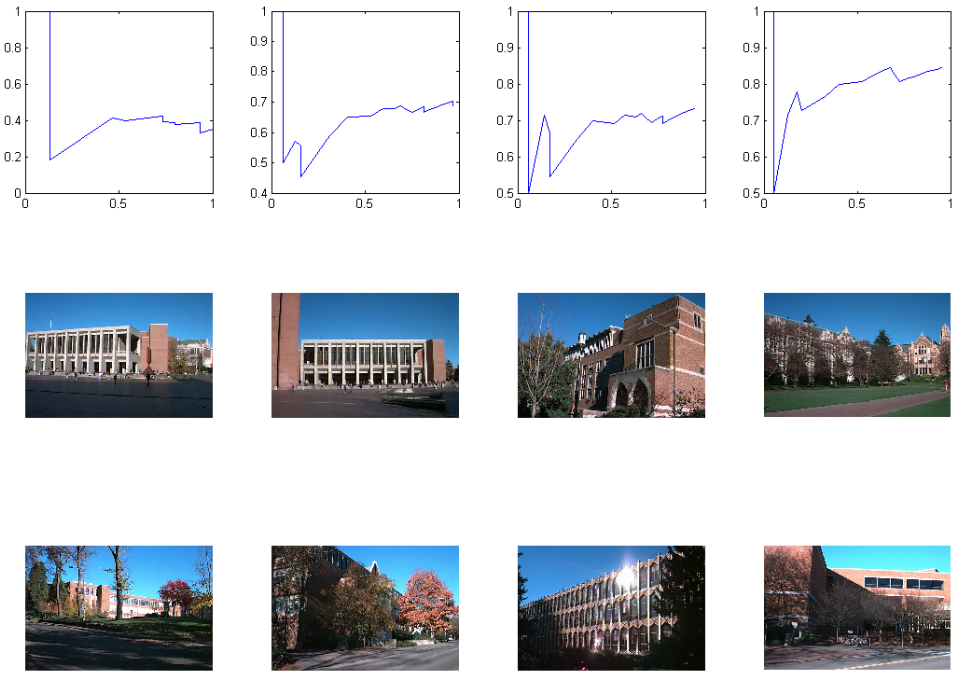


Fig. 10. The ranked retrieved images and PR graphs for the SCD

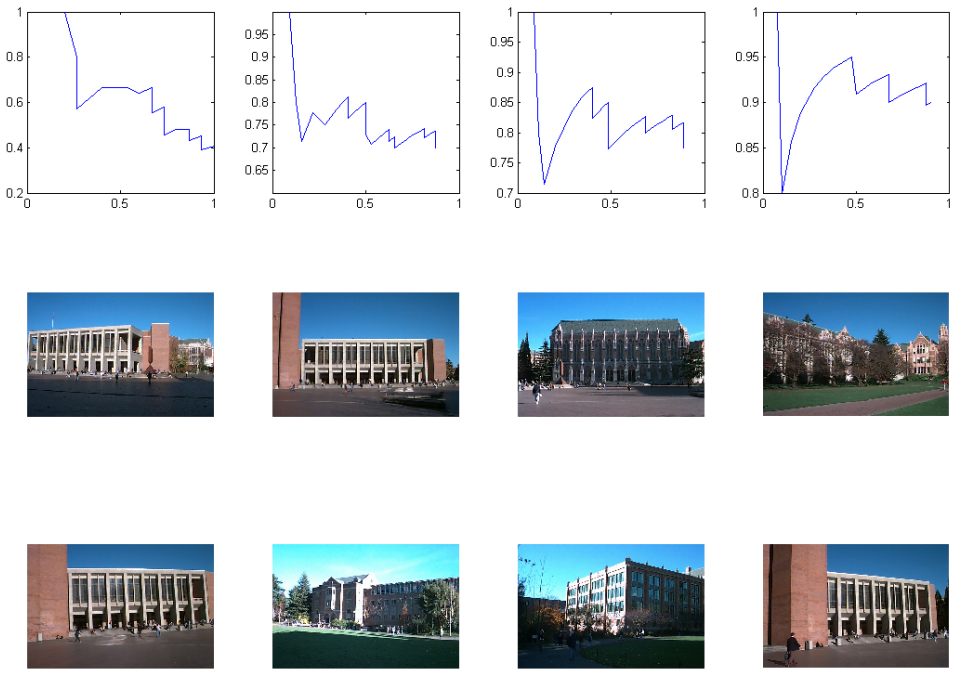


Fig. 11. The ranked retrieved images and PR graphs for the EHD

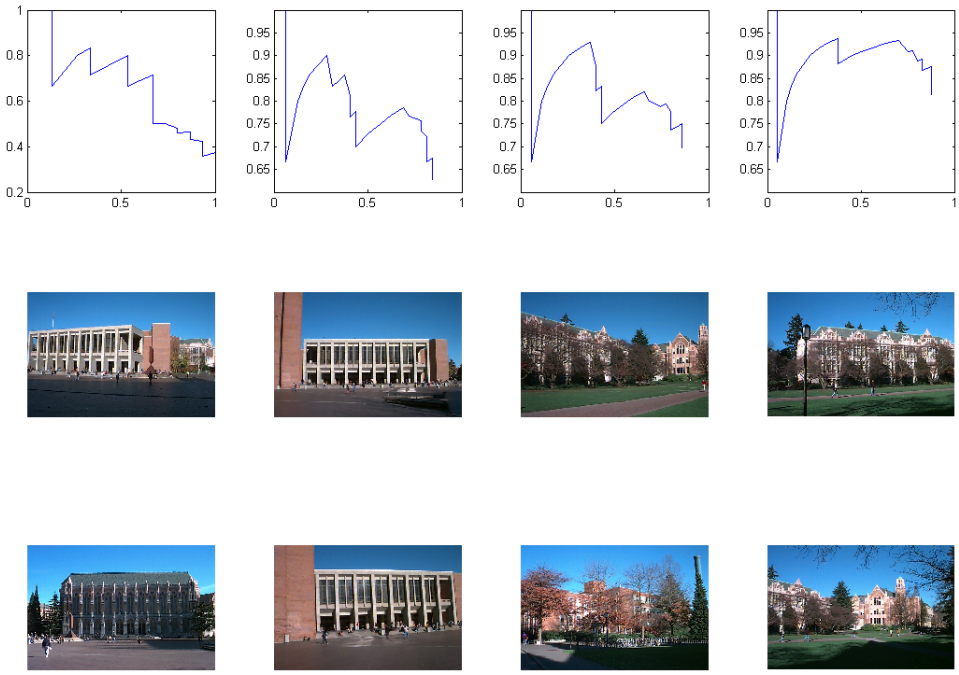


Fig. 12. The ranked retrieved images and PR graphs for combined descriptors

A. Initial Tests

The descriptors for the original images were compared to the descriptors for the 'lossless' scaled versions to see if the images had been modified significantly. As shown in table IX and table X, the query descriptor differs from the descriptor of the original image from which it was derived. If the image had been identical to the original after being processed by the emulator the L_2Norm would have been zero, as in the non-scalable tests. The L_2Norm for the query image is significantly lower than those of the other images in the set. Therefore it is still returned as the top result. This is not always the case when the resolution and quality are reduced.

As shown in figures 13 and 14, the images ranked 2nd and 3rd exchange places in the top five results for 'Image04'.

Fig. 13. Combined descriptors for original Image04

Fig. 14. Combined descriptors for resized 'lossless' Image04

L2Norms	5.3037	13.7664	13.8285	15.6937	16.5724
Image numbers	4	31	19	36	35

TABLE IX

THE L_2Norm VALUES FOR A RESIZED IMAGE04 USING COMBINED DESCRIPTORS

As shown in figure 15 and figure 16, in the top five results for 'Image06' the images ranked 4th and 5th exchange places.

Fig. 15. Combined descriptors for original Image06

Fig. 16. Combined descriptors for resized 'lossless' Image06

B. CLD

1) Lowest quality and resolution: The images returned when using only the CLD and lowest quality and resolution descriptors are shown in figure 17. All components of the CLD are equally weighted. The resolution setting is for sixteenth size (2) and the quality setting is for poor (8).

Fig. 17. The retrieved images for Image06 with the CLD and lowest quality and resolution

2) Medium quality and resolution: The results for the CLD, when the resolution setting is for quarter size (1) and the quality setting is for medium (5), are presented in figures 19 and 20, and tables XIII and XIV.

L2Norms	5.3619	14.4786	15.6501	16.3821	16.464
Image numbers	6	5	9	29	10

TABLE X

THE L_2Norm VALUES FOR A RESIZED IMAGE06 USING COMBINED DESCRIPTORS

L2Norms	1.5159	5.4999	6.1395	6.533	6.6148
Image numbers	6	10	29	17	5

TABLE XI

THE RANKINGS FOR IMAGE06 WITH THE CLD AND LOWEST QUALITY AND RESOLUTION

Fig. 18. The retrieved images for Image04 with the CLD and lowest quality and resolution

L2Norms	1.0696	6.2408	6.3479	6.4829	6.4985
Image numbers	4	45	48	35	19

TABLE XII

THE RANKINGS FOR IMAGE04 WITH THE CLD AND LOWEST QUALITY AND RESOLUTION

Fig. 19. The retrieved images for Image06 with the CLD and medium quality and resolution

L2Norms	0.85771	5.6571	6.2505	6.6182	6.8
Image numbers	6	10	29	17	5

TABLE XIII

THE RANKINGS FOR IMAGE06 WITH THE CLD AND MEDIUM QUALITY AND RESOLUTION

Fig. 20. The retrieved images for Image04 with the CLD and medium quality and resolution

L2Norms	1.0637	6.2037	6.5456	6.5968	6.6607
Image numbers	4	45	48	31	35

TABLE XIV

THE RANKINGS FOR IMAGE04 WITH THE CLD AND MEDIUM QUALITY AND RESOLUTION

C. SCD

The SCD can be truncated, as the Haar transform puts the most significant coefficients at the beginning of the descriptor [1]. Figure 21 shows the L_2Norm and L_1Norm values when using only the 1st eight SCD coefficients. Four images have been tested and averaged. See appendix A for the individual values.

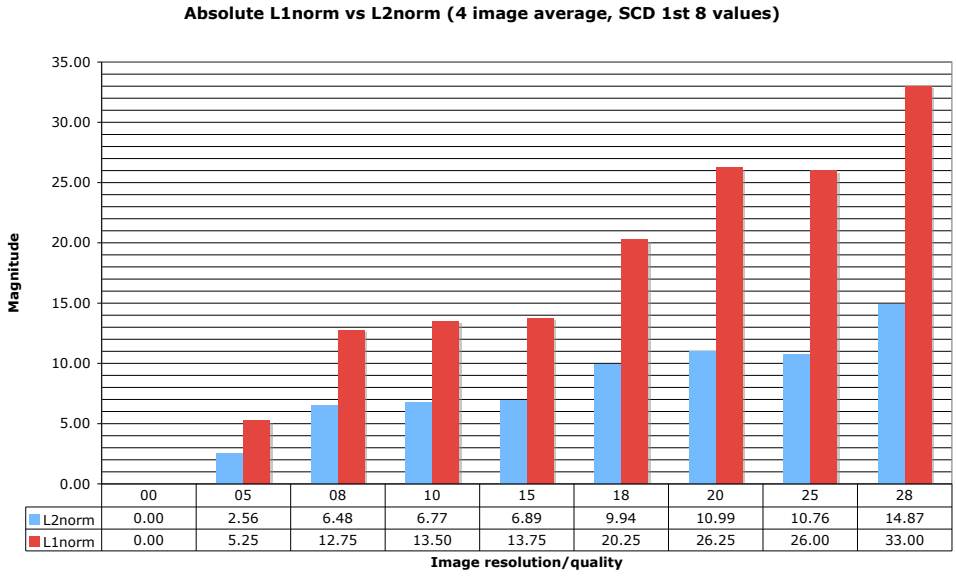


Fig. 21. The 1st 8 coefficients of the SCD for an average of 4 images, L_2Norm and L_1Norm

1) Lowest quality and resolution: The results for the SCD, when the resolution setting is for sixteenth size (2) and the quality setting is for low (8), are presented in figures 22 and 23, and tables XV and XVI.

2) Medium quality and resolution: The results for the SCD, when the resolution setting is for quarter size (1) and the quality setting is for medium (5), are presented in figures 24 and 25, and tables XVII and XVIII.

Fig. 22. The retrieved images for Image06 with the SCD and lowest quality and resolution

L2Norms	38.6652	39.975	42.4971	43.1741	43.3128
Image numbers	6	5	8	19	10

TABLE XV

THE RANKINGS FOR IMAGE06 WITH THE SCD AND LOWEST QUALITY AND RESOLUTION

Fig. 23. The retrieved images for Image04 with the SCD and lowest quality and resolution

L2Norms	38.2884	41.821	42.0595	43.7836	44.0227
Image numbers	8	15	14	4	5

TABLE XVI

THE RANKINGS FOR IMAGE04 WITH THE SCD AND LOWEST QUALITY AND RESOLUTION

Fig. 24. The retrieved images for Image06 with the SCD and medium quality and resolution

L2Norms	26.8514	35.0999	36.6879	37.8021	39.281
Image numbers	6	5	10	9	29

TABLE XVII

THE RANKINGS FOR IMAGE06 WITH THE SCD AND MEDIUM QUALITY AND RESOLUTION

Fig. 25. The retrieved images for Image04 with the SCD and medium quality and resolution

L2Norms	23.9583	31.686	32.9393	33.6006	33.9116
Image numbers	4	14	26	8	31

TABLE XVIII

THE RANKINGS FOR IMAGE04 WITH THE SCD AND MEDIUM QUALITY AND RESOLUTION

D. EHD

1) Lowest quality and resolution: The results for the EHD, when the resolution setting is for sixteenth size (2) and the quality setting is for low (8), are presented in figures 26 and 27, and tables XIX and XX.

Fig. 26. The retrieved images for Image06 with the EHD and lowest quality and resolution

L2Norms	15.1658	16.1245	17.1172	17.4642	18.3848
Image numbers	6	36	31	46	19

TABLE XIX

THE RANKINGS FOR IMAGE06 WITH THE EHD AND LOWEST QUALITY AND RESOLUTION

2) Medium quality and resolution: The results for the EHD, when the resolution setting is for quarter size (1) and the quality setting is for medium (5), are presented in figures 28 and 29, and tables XXI and XXII.

Fig. 27. The retrieved images for Image04 with the EHD and lowest quality and resolution

L2Norms	10.2956	10.8628	11.9164	12.0416	12.49
Image numbers	46	4	24	36	31

TABLE XX

THE RANKINGS FOR IMAGE04 WITH THE EHD AND LOWEST QUALITY AND RESOLUTION

Fig. 28. The retrieved images for Image06 with the EHD and medium quality and resolution

L2Norms	12.2882	15.7797	15.8114	15.843	16.0624
Image numbers	6	36	31	19	4

TABLE XXI

THE RANKINGS FOR IMAGE06 WITH THE EHD AND MEDIUM QUALITY AND RESOLUTION

Fig. 29. The retrieved images for Image04 with the EHD and medium quality and resolution

L2Norms	10.5357	13.0384	13.4536	13.8203	13.8203
Image numbers	4	36	18	24	35

TABLE XXII

THE RANKINGS FOR IMAGE04 WITH THE EHD AND MEDIUM QUALITY AND RESOLUTION

E. Combined Descriptors

The experiments were repeated with the descriptors combined. A weighting of 4 for the SCD, 2 for the CLD luminance, 1 for both CLD chrominance and 8 for the EHD was used.

Fig. 30. The retrieved images for Image06 with the lowest quality and resolution

1) Lowest quality and resolution: Table XXIII shows that the L_2Norm values are different to table X when the resolution and quality are reduced to the lowest settings used. Four out of the top five image ranks returned remain unchanged for 'Image06' when compared to the 'lossless' image in table X. The 3rd result returned is different but still looks similar to the other results.

L2Norms	14.3766	18.8162	19.7967	19.9162	19.9412
Image numbers	6	5	7	29	10

TABLE XXIII

THE RANKINGS FOR IMAGE06 WITH THE LOWEST QUALITY AND RESOLUTION

Fig. 31. The retrieved images for Image04 with the lowest quality and resolution

Table XXIV and figure 31 show that although the L_2Norm values are different to table IX when the resolution and quality are reduced to the lowest settings, the top four image ranks returned remain unchanged for 'Image04' from table IX.

2) Medium quality and resolution: When the descriptors from images re-coded with medium resolution and quality (1 and 5 respectively) were used, the following results were observed. Figure 32 and table XXV when compared to table X show that the top five images are the same, but in a different order. The same is also true of 'Image04' when comparing table IX with table XXVI.

L2Norms	13.5452	17.1842	17.6768	17.8847	17.9037
Image numbers	4	31	19	36	46

TABLE XXIV

THE RANKINGS FOR IMAGE04 WITH THE LOWEST QUALITY AND RESOLUTION

Fig. 32. The retrieved images for Image06 with the medium quality and resolution

L2Norms	10.5929	16.8659	17.4709	17.5757	18.4481
Image numbers	6	5	29	10	9

TABLE XXV

THE RANKINGS FOR IMAGE06 WITH THE MEDIUM QUALITY AND RESOLUTION

Fig. 33. The retrieved images for Image04 with the medium quality and resolution

L2Norms	9.4368	15.652	15.6689	16.6433	17.0615
Image numbers	4	19	31	36	35

TABLE XXVI

THE RANKINGS FOR IMAGE04 WITH THE MEDIUM QUALITY AND RESOLUTION

F. Average L_2Norms

Experiments were done to see if the trends observed in one image set were applicable to images with a different subject matter. One image was chosen from each set in the database. Some sets could not be used for technical reasons. The average difference in L_2Norm values were calculated and plotted for nine images, see figure 34. The numerical data is presented in appendix E.

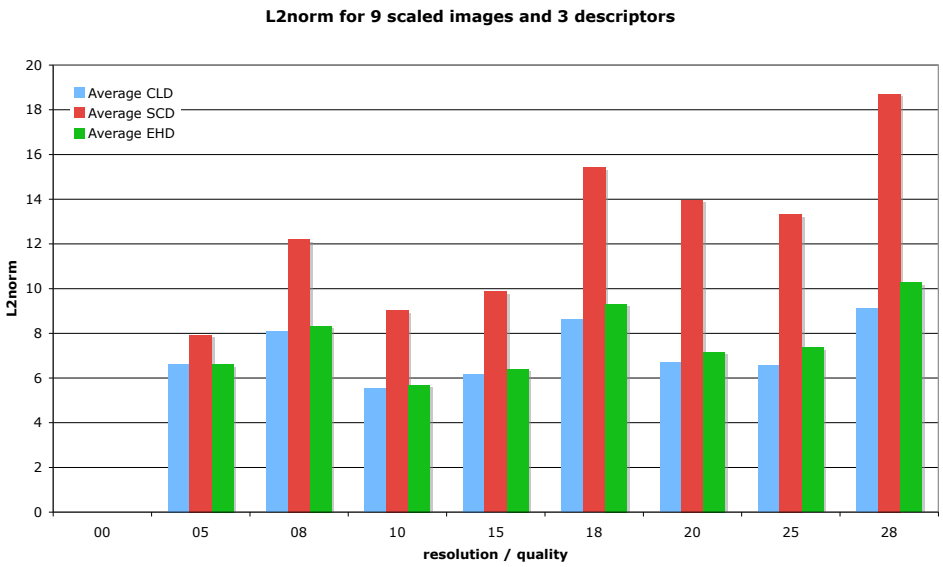


Fig. 34. A graph of the average difference in L_2Norm values for scaled images

G. Performance

The following figures show the PR graphs and top eight matches for the two test images and three quality settings used throughout the rest of the results. The rightmost PR graph is when the greatest number of images are deemed relevant and for the leftmost the least are. Recall is on the x axis and precision is on the y axis. The images are ranked 1st, 2nd, 3rd and 4th from left to right respectively on the top line, and similarly from 5th through 8th on the bottom.

Figures 35 and 36 show the retrieval performance for full quality and resolution. Figures 37 and 38 show the retrieval performance for medium quality and resolution. Settings 5 and 1 (quarter resolution) respectively. Figures 39 and 40 show the retrieval performance for low quality and resolution. Settings 8 and 2 (sixteenth resolution) respectively.

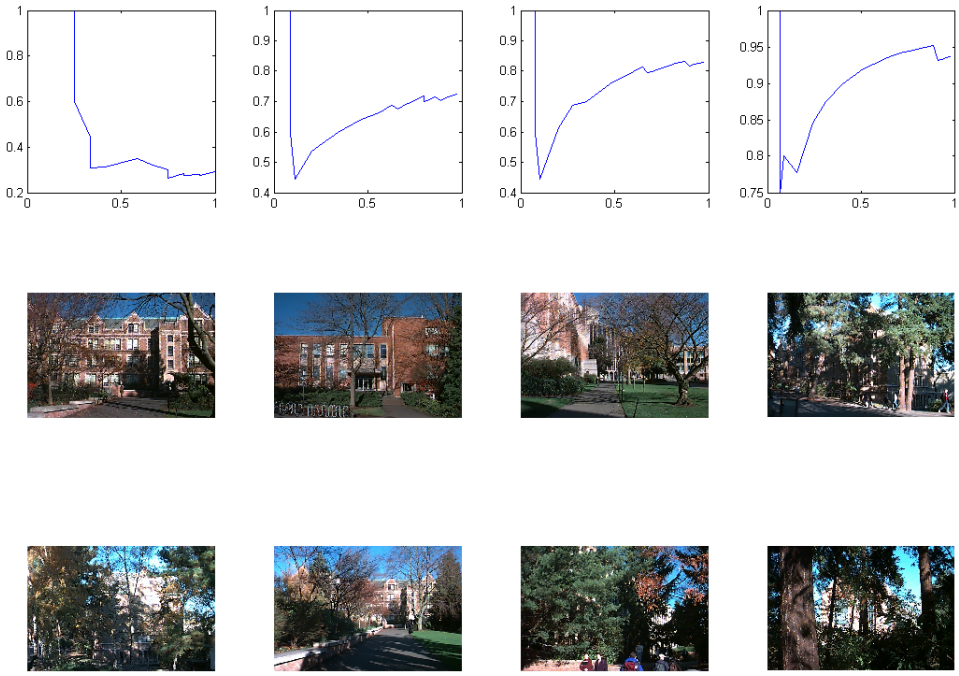


Fig. 35. Image04 retrieval, combined descriptors, full quality and res.

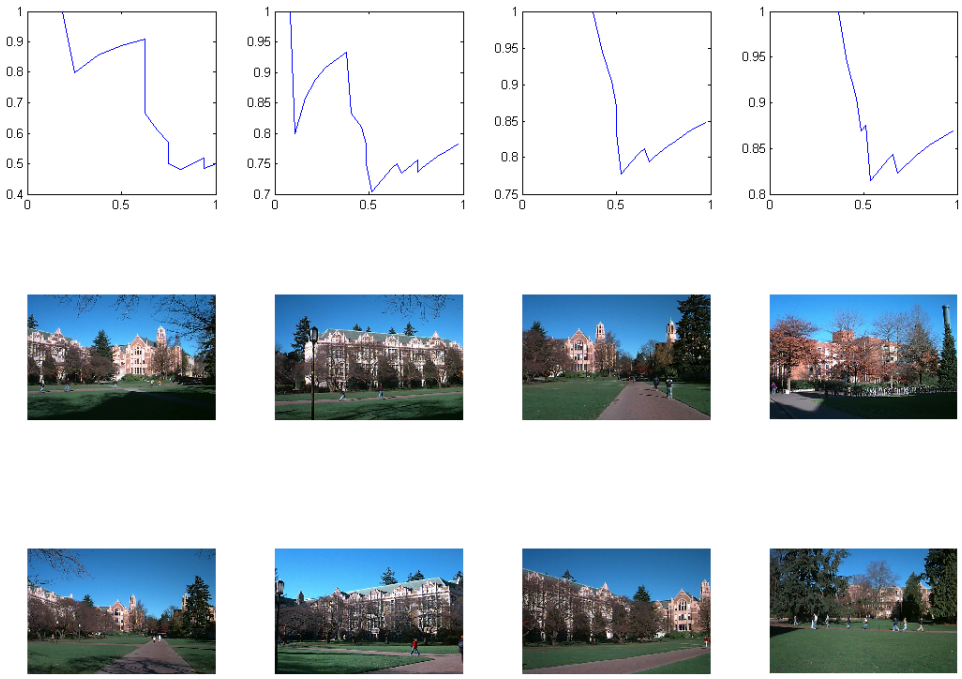


Fig. 36. Image06 retrieval, combined descriptors, full quality and res.

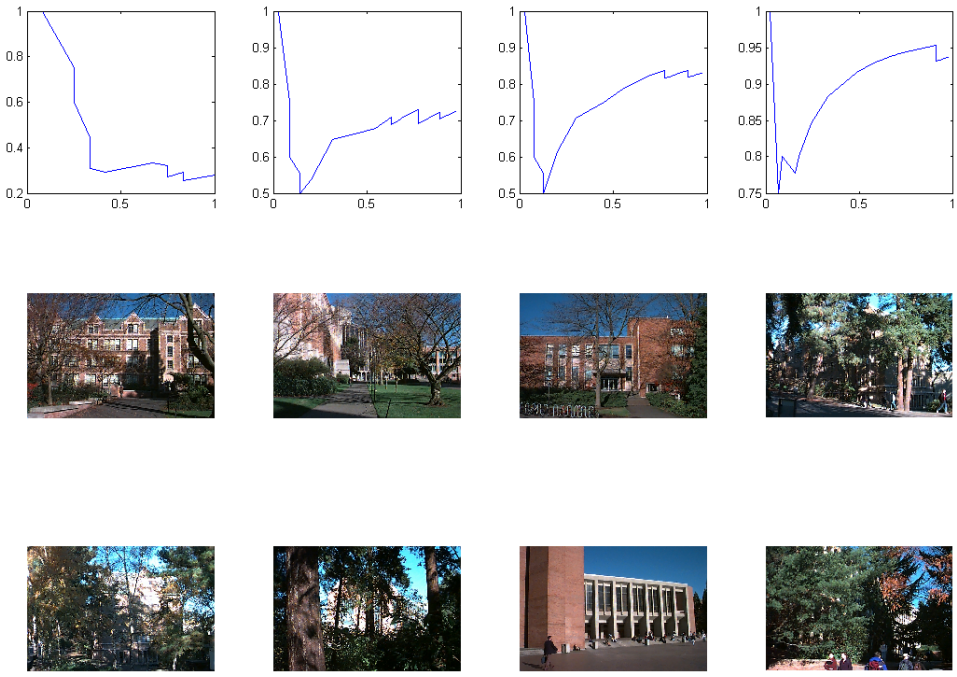


Fig. 37. Image04 retrieval, combined descriptors, medium quality and res.

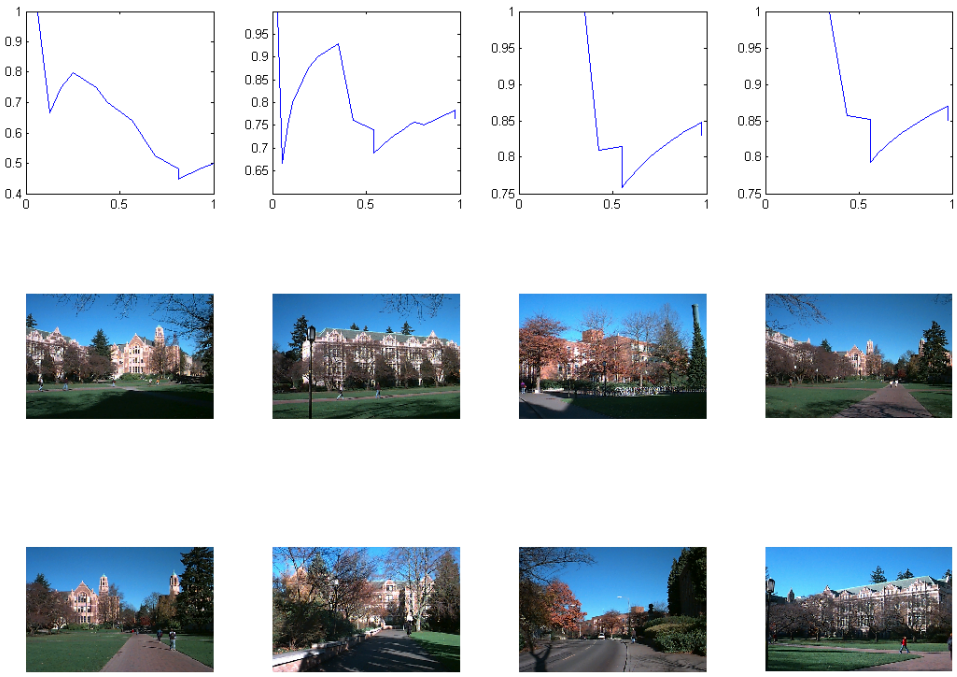


Fig. 38. Image06 retrieval, combined descriptors, medium quality and res.

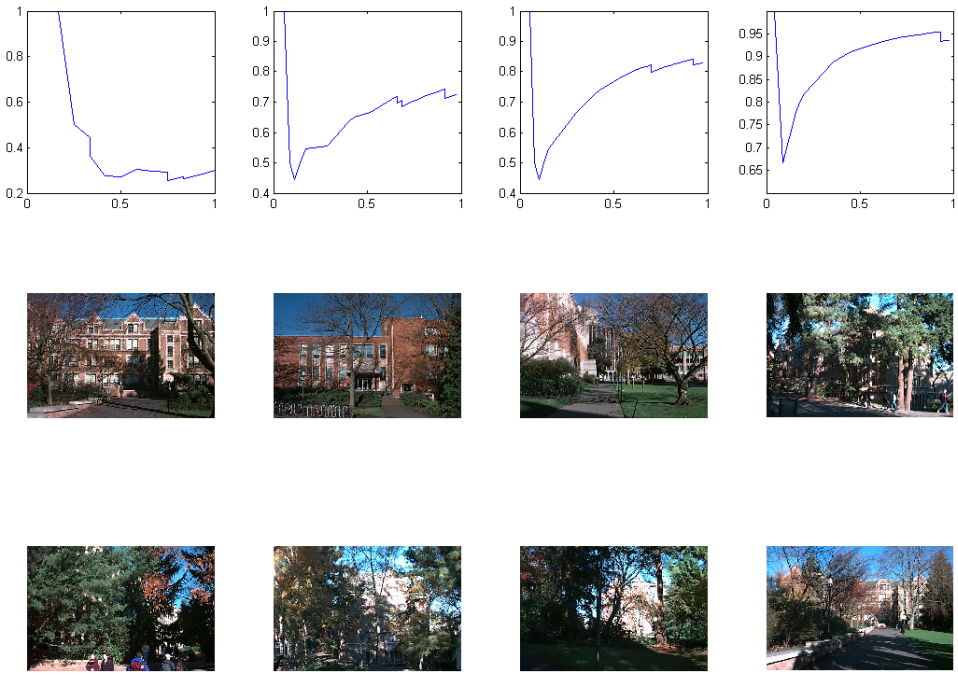


Fig. 39. Image04 retrieval, combined descriptors, low quality and res.

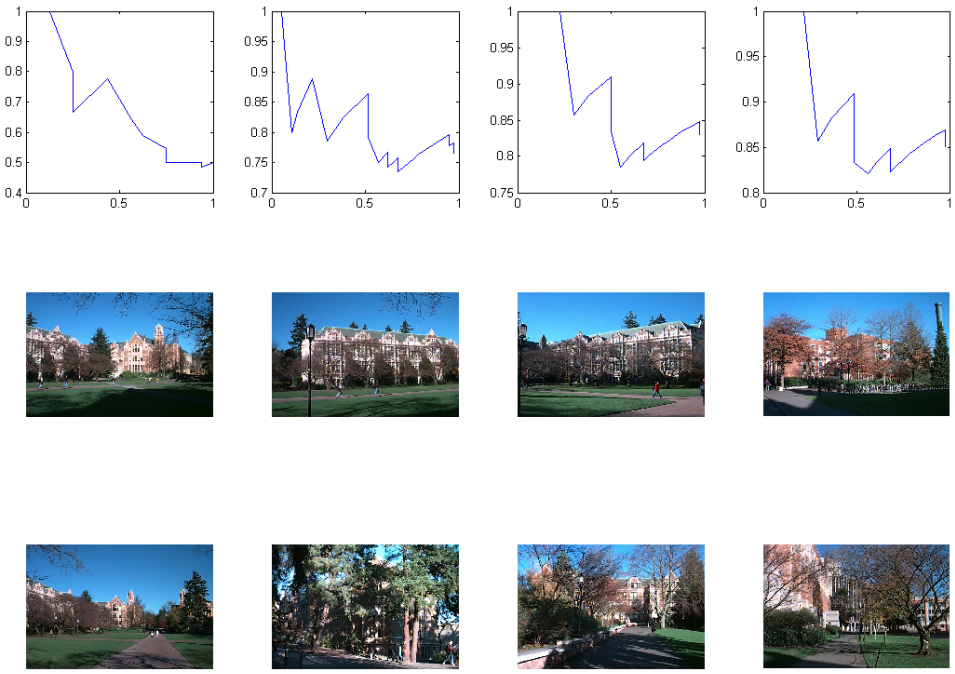


Fig. 40. Image06 retrieval, combined descriptors, low quality and res.

VIII. Discussion

The software chosen performed well. CALIPH could be extended as it is open source or LIRE could be used in a program which incorporates scalable images.

Storing the MPEG-7 descriptors as XML files was a good decision as it allowed easy analysis. Example files are shown in appendix K.

A. Retrieval

The system performed reasonably well at retrieving images, although it is limited by the number and nature of the descriptors used. To the human eye the images that were returned by the matching algorithm looked similar to the query image. This is positive, as the ultimate aim of such a CBIR system would be to satisfy a human user.

The CLD produced good results, although the dynamic range of the values was low. In other words, the L_2Norm values for the similar images were close together and hard to distinguish.

The correct weighting for the six different components was hard to determine. Human perception of colour is not as good as for that of brightness, so logically the chrominance should have a lower weighting than the luminance. Using a similar weighting to existing video standards seemed to produce good results. Each chrominance is given half the significance of the luminance.

The SCD also produced good results in retrieval. It subjectively appeared to return more images of the same scene than the CLD. The SCD is by nature scalable in that it can be truncated, but this does not necessarily mean that it performs well with scalable images.

The EHD appeared to work well with images of artificial objects, e.g. buildings, but not so well with images of natural objects, e.g. trees. Images with many straight lines were matched fairly accurately but images with more complex shapes were not.

Figure 12 shows how the performance is improved when the descriptors are combined. Precision is higher than those of each individual descriptor as demonstrated by the PR graphs.

The PR graphs produced in the experiments did not always exhibit a smooth inverse relationship. They were often jagged. Perhaps this was due to the limited nature of the ground truth data provided. There were only a small number of levels that could be used.

The image display component worked well and is shown in appendix H. It can be customised to allow any number of images, although the size of the images will get smaller as more images are displayed. This is why a maximum of five images on one line was used. The generation of latex tables was also useful. When the performance evaluation was combined with the image display then lots of information could be displayed graphically together.

B. Scalability

The results of the experiments on scalable images indicate that the descriptors do not vary much as the resolution and quality is decreased. The different amounts of similarity that each descriptor exhibited is likely to be due to the different ways that they are calculated.

Re-scaling the images with the JPEG2000 emulator modified them even if the full resolution and quality version was generated. This could be down to the resizing of the images that had to take place or due to the WT itself. Alternative methods could be used, such as cropping the images or adding black bands. Reducing the number of wavelet decomposition levels would also help, as the image dimensions have to be devisable by 2^{levels}. When padding the images to the required size by adding black bands was tried, it altered the descriptors more than resizing the images. The CLD was changed significantly from the original version, which made sense, as the image looked different.

The CLD produced the most scaleable results and seemed to be the most resolution independent of all the descriptors tested. The coefficients differed least from the original versions of the images. This was expected due to the way that it breaks the image into an 8×8 block of colours [1]. Tables XI, XII, XIII and XIV indicate that the L_2Norm for the query image is very low, i.e. the differences between the original image and the low and medium versions of it that have been processed by the emulator are small, as far as the CLD is concerned. When compared to the full resolution and quality version generated by the emulator instead of the original image, the CLD performance is less distinguishable from that of the EHD.

When matching descriptors from the low resolution and quality images, the SCD did not perform well. When comparing table XV and V, only the top two images are the same. The results are even worse for 'Image04'. The query image is ranked 4th in table XVI and the top match of 'Image08' does not appear in table VI.

In figure 34 and appendix E it can be seen that the SCD produces much higher L_2Norm values than the other two descriptors, although it follows the same trend as them. It can also be seen in figure 21 and appendix A, that the SCD can be truncated to only eight coefficients and it still shows the same behaviour, although the results are not as accurate.

On the two test images, the EHD did not perform so well when presented with the low resolution and quality images. The L_2Norm for the query image was high and close in value to the other top results returned. For a query of 'Image04' the results were so similar that the query image was not returned as the top result, but came 2nd.

When tested at the medium quality and resolution, the EHD faired a little better, although it did not return the same results as when tested on the non-scalable images.

Figure 34 shows that although the EHD performed much worse than the CLD when matching

images to the original unmodified versions, the differences between them for different qualities and resolutions were small. This suggests that the process of re-scaling the image has a large effect on the EHD but little effect on the CLD.

All three descriptors seem to vary the same as quality and resolution are reduced, but this does not correspond to similar performance in matching behaviour. The process of performing the JPEG2000 emulation alters the performance of each descriptor by very different amounts. The SCD and EHD are affected greatly, whilst the CLD seems largely immune.

The results show that an efficient CBIR system could be built that only used part of the data from images. This would reduce the requirements on bandwidth and storage for such a system. The whole image file would not need to be processed, but the images would have to be in a scalable format such as JPEG2000 to have maximum benefit. This could speed up the retrieval process, especially on low powered hardware.

IX. Conclusions

The aims and specification for the project have been met. A CBIR system has been built and evaluated. It was then used to experiment on scalable images and the results were documented.

The CBIR system that was built was modular in nature. It consisted of indexing, retrieval and evaluation components. For the indexing, various pieces of software were researched and a suitable one was chosen. The retrieval and evaluation components were programmed after learning the Java programming language. The scalability and display components were used and built respectively after learning how to use MATLAB. Performance evaluation metrics were calculated for images that had been matched.

It has been discovered that not all of the image data is required in order to extract descriptors that achieve an acceptable match. Each of the descriptors experimented upon perform differently when presented with scaled images. The CLD was found to be the best descriptor for tolerance to changes in resolution and quality.

X. Recommendations

The performance of the code could be improved to make the retrieval work faster. Java is not a very fast language when programs are run but software development can be done quickly. If the system was re-written in the C programming language or Assembly language it would perform faster. The main bottle neck was opening and reading the many XML files. If a different format, such as a binary file was used, then the speed would be increased, although the readability of XML would be lost. The file accesses could be buffered and optimised as some files are read more than once. The system is only a proof of concept prototype and not a production system, so speed was not important.

More versions of an image could be generated and used, e.g. five resolutions and ten quality settings. If higher resolution images were used to start with then a greater number of smaller resolutions could be generated before they become too small to be useful. It would be convenient to have all the images be compatible with the JPEG2000 emulation software originally. It could then be guaranteed that any undesired modification of the images, and hence the descriptors, would be a result of the WT and not the initial resizing of the images.

More experiments could be done with larger data sets. They might discover wider trends. It may also be possible to determine the ideal weightings of the descriptors, as the weightings used in the results may not be the best for all types of images. A large, statistically valid ground truth would be needed. This would have to use a lot of people to describe images but could be done online.

The results suggest that there is a relationship between the reduction in quality and resolution of the images and the reduction in matching capability of the MPEG-7 descriptors. This may be able to be proved mathematically, but the modification of the images by the JPEG2000 emulation software would have to be taken into account. If a complete system were to be designed from scratch then it may be possible to engineer it so that the undesired modification of the images is minimised.

More experiments could be done in an attempt to discover the extent to which the JPEG2000 emulation software and required resizing modify the images. The results could be skewed by this and if it were better understood or eliminated then the results would have a greater significance.

Extensions to MPEG-7 could be developed that include scalable images. The standard can be expanded and new descriptors are allowed to be written and added [1].

It is felt that there is much scope to expand upon the work of this project. Some very interesting and potentially useful results have been discovered that demand further investigation.

REFERENCES

[1] B.S. Manjunath, P.P. Salembier, and T. Sikora, *Introduction to MPEG-7: Multimedia Content Description Interface*, John Wiley and Sons, 2002.

[2] B.S. Manjunath, "Color and texture descriptors," *IEEE Transactions on circuits and systems for video technology*, vol. 11, no. 6, 2001.

[3] K.M. Wong, K.W. Cheung, and L.M. Po, "MIRROR: an interactive content based image retrieval system," *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*, pp. 1541–1544, 2005.

[4] M. Gabbouj, S. Kiranyaz, K. Caglar, B. Cramariuc, F.A. Cheikh, O. Guldogan, and E. Karaoglu, "MUVIS: A Multimedia Browsing, Indexing and Retrieval System," *Proceedings of the IWDC 2002 Conference on Advanced Methods for Multimedia Signal Processing*, 2002.

[5] J.R. Smith and S.F. Chang, "VisualSEEk: a fully automated content-based image query system," *Proceedings of the fourth ACM international conference on Multimedia*, pp. 87–98, 1997.

[6] L. Gagnon, "R&D status of ERIC-7 and MADIS: two systems for MPEG-7 indexing/search of audio-visual content," *Proc. SPIE Conference on Multimedia Systems and Applications*, vol. 8, pp. 12, 2005.

[7] A. Khella and B.B. Bederson, "Pocket PhotoMesa: a Zoomable image browser for PDAs," *Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia*, pp. 19–24, 2004.

[8] T. Sikora, "The MPEG-7 visual standard for content description-an overview," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 11, no. 6, pp. 696–702, 2001.

[9] M. Lux, W. Klieber, and M. Granitzer, "Caliph & Emir: Semantics in Multimedia Retrieval and Annotation," *19th CODATA Conference, Berlin, Germany*, 2004.

[10] M. Lux, J. Becker, and H. Krottmaier, "Caliph & Emir: Semantic Annotation and Retrieval in Personal Digital Photo Libraries," *15th Conference on Advanced Information Systems Engineering (CAiSE), Velden, Austria*, 2003.

[11] LG Shapiro, "Object and Concept Recognition For Content Based Image Retrieval, Ground Truth Database," *Report Nr. IIS-0097329, Department of Computer Science and Engineering, University Of Washington, Seattle, WA, USA*, 2004.

[12] Y. Li and L.G. Shapiro, "Consistent line clusters for building recognition in CBIR," *Proceedings of the International Conference on Pattern Recognition*, pp. 952–956, 2002.

[13] H. Ding, "Compressed domain analysis of scalable coded video content," M.S. thesis, University of Sheffield, 2006.

[14] J. Li and JZ Wang, "Automatic Linguistic Indexing of Pictures by a statistical modeling approach," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 9, pp. 1075–1088, 2003.

[15] JZ Wang, J. Li, and G. Wiederhold, "SIMPLIcity: semantics-sensitive integrated matching for picturelibraries," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 9, pp. 947–963, 2001.

[16] M. Sanderson, *Word sense disambiguation and information retrieval*, Ph.D. thesis, University of Glasgow, 1996.

[17] G. Park, Y. Baek, and H.K. Lee, "A ranking algorithm using dynamic clustering for content-based image retrieval," *International Conference on Image and Video Retrieval, Lecture Notes in Computer Science*, vol. 2383, pp. 316–324, 2002.

APPENDIX

A. Matching scheme comparison

L2Norm					L1Norm of absolute values				
im2	im3	im4	im5	average	im2	im3	im4	im5	average
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6.78	0.00	1.00	2.45	2.56	16.00	0.00	1.00	4.00	5.25
7.87	1.00	6.00	11.05	6.48	18.00	1.00	10.00	22.00	12.75
6.48	6.16	7.21	7.21	6.77	14.00	12.00	14.00	14.00	13.50
5.39	7.68	7.00	7.48	6.89	13.00	13.00	13.00	16.00	13.75
9.17	7.62	9.43	13.56	9.94	20.00	16.00	15.00	30.00	20.25
9.59	14.14	10.68	9.54	10.99	22.00	34.00	24.00	25.00	26.25
8.25	14.73	10.95	9.11	10.76	20.00	37.00	24.00	23.00	26.00
14.04	13.82	14.76	16.85	14.87	33.00	29.00	32.00	38.00	33.00

B. Matching component source code

```
/*
MPEG-7 Scalable CBIR Project: Matching component
James Singleton (C) 2007
*/

import javax.xml.xpath.*;
import javax.xml.parsers.*;
import org.w3c.dom.*;

public class l2norm
{
    public static void main(String[] args)
    throws ParserConfigurationException, XPathExpressionException, org.xml.sax.SAXException,
java.io.IOException
    {
        int numoffiles = 48; //number of files in the image set. Could be taken from argument.
        String[] xmlfiles = new String[numoffiles];
        for(int i = 0; i < numoffiles; i++)
            {
            String temp = Integer.toString(i+1);
            if (temp.length() == 1)temp = '0' + temp;
            xmlfiles[i] = "Image" + temp + ".mp7.xml";
            }
    //Display usage information if no argument is used
    if(args.length == 0)
    {
    System.out.print('\n');
    System.out.println("Usage: java l2norm [filename]");
    System.out.println("The arguments must contain the filename of the reference mp7 xml file.");
    System.out.println("Output suitable for redirection to a spreadsheet (tab separted values).");
    System.out.println("E.g. java l2norm image1.mp7.xml > image.xls");
    System.out.print('\n');
    //for(String tempz: files)System.out.println(tempz);//testing filenames array
    System.exit(1);
    }
        //Reference data file to which all others are compared, from 1st argument
        String queryxmlfile =  args[0];
        //Arrary of the strings to access the required data in the xml file for each descriptor
        String[] desciptXpaths = { //Xpaths to locate the descriptor's tags in the XML files
        "//VisualDescriptor/Coeff/text()",  //SCD
        "//VisualDescriptor/YDCCoeff",      //CLD x 6 Y, Cb and Cr (AC and DC)
        "//VisualDescriptor/YACCoeff63",
        "//VisualDescriptor/CrDCCoeff",
        "//VisualDescriptor/CrACCoeff63",
        "//VisualDescriptor/CbDCCoeff",
        "//VisualDescriptor/CbACCoeff63",
        "//VisualDescriptor/BinCounts",     //EHD
        };
        //double[] descWeightings ={0,1,1,1,1,1,1,0,};//Weightings for each descriptor.CLD equal
        //double[] descWeightings ={1,0,0,0,0,0,0,0,};//Weightings for each descriptor.SCD
        //double[] descWeightings ={0,0,0,0,0,0,0,1,};//Weightings for each descriptor.EHD
          double[] descWeightings ={4,2,2,1,1,1,1,8,};//Weightings for each descriptor.Combination
        //BUILD THE DATA
        double[][] L2s = new double[xmlfiles.length][desciptXpaths.length + 1];
        //2D array to store results
        for(int count2 = 0 ; count2 < desciptXpaths.length ; count2++)
        //Iterate through the descriptors
        {
            String tagstring = GetXmlFiles(queryxmlfile,desciptXpaths[count2]);
            //get the data for a descriptor from the reference file
            //System.out.println(tagstring);
            for(int count = 0 ; count < xmlfiles.length ; count++) //Iterate through the files
            {
                String tagstring2 = GetXmlFiles(xmlfiles[count],desciptXpaths[count2]);
                //get the data for a descriptor from a file
                //System.out.println(tagstring2);
                //for(double tmp:StringtoArray(tagstring2))System.out.print(tmp + "\t");
                //System.out.print('\n');
                //Compute the L2Norm of the same descriptor for 2 files (reference and sample)
                //store in 2D array
                L2s[count][count2] =
                L2NormArrays(StringtoArray(tagstring),StringtoArray(tagstring2));
                //System.out.println(L2s[count][count2]); //Print values. Testing only
            }
        }
```

```
        //INTERMEDIATE TESTING OF VALUES
        //for(double tmp : descWeightings)System.out.print(tmp + ":");
        //System.out.print('\t');
        double TotalWeightings = 0.0; //Used to calculate the average later, keep at zero
        for(double temptot:descWeightings) TotalWeightings += temptot;
        //Calculate Total from weightings array
        //System.out.println(TotalWeightings);

        //SHOW THE DATA
        for(int count = 0 ; count < xmlfiles.length ; count++)
        //Display the results to stdout for each file
        {
        //Ajust Weighting here. Acc L2n with bias. (from array)
            double Sum = 0.0; // tempary variable to store the sum of L2Norms for each descriptor
            for(int count2 = 0 ; count2 < desciptXpaths.length ; count2++)
            //cycle through the descriptors
            {
                //System.out.println(L2s[count][count2]);
                Sum += (L2s[count][count2])*descWeightings[count2]; //Accumulate the results.
        }
            //System.out.println(Sum);// Test intermediate value.
            L2s[count][desciptXpaths.length] = Sum/TotalWeightings;
            //Store average L2N in another element.
            //System.out.println(xmlfiles[count]);
            System.out.print(L2s[count][desciptXpaths.length]);//print the value
            System.out.print('\t');//tabs to separate the values
            //System.out.print('\n');
        }
    System.out.print('\n');//newline to end the entry
    }
    public static double L2NormArrays(int[] a, int[] b)
    //Computes the L2norm, also includes some error handleing.
    {
        if(a.length == b.length)
        {
            double l2n = 0.0;
            for(int i = 0 ; i < a.length ; i++)l2n += Math.pow(a[i]-b[i],2);
            //raise difference to power 2 and acc.
            //System.out.println(l2n); // Test intermediate value.
            l2n = Math.sqrt(l2n);// works out the l2norm, could be modified to support lpnorm
            return l2n;
        }
        else
        {
            System.out.println("Arrays not of equal length");
            return -1.0;
            //Should never happen
        }
    }
    public static int[] StringtoArray(String ints)
    {
        ints = ints.trim();//trims whitespce to prevent certain errors occuring in the next step.
        String temp[] = ints.split("\\s");
        //Tokenizes (splits) the string into an array of strings. No whitespace.
        int tmplngth = temp.length;
        //Allows only a subset of the data to be used, e.g. only the 1st 16 coefficients
        int[] result = new int[tmplngth]; //Init to size of split string array
        //for(String n:temp)System.out.println(Integer.parseInt(n));
        for(int i=0; i<tmplngth; i++)result[i] = Integer.parseInt(temp[i]);
        return result;
    }
    public static String GetXmlFiles(String documentName, String expression)
    throws ParserConfigurationException, XPathExpressionException, org.xml.sax.SAXException,
java.io.IOException
        {
            DocumentBuilder parser = DocumentBuilderFactory.newInstance().newDocumentBuilder();
            Document doc = parser.parse(new java.io.File(documentName));
            XPath xpath = XPathFactory.newInstance().newXPath();
            return(xpath.evaluate(expression, doc));
        }
}
```

C. Ground truth evaluation source code

```
/*
MPEG-7 Scalable CBIR Project: Ground truth evaluation component
A Java program to calculate a numerical ground truth from a textual version for a set of images
James Singleton (C) 2007
*/

import java.io.*; //for file I/O
import java.util.*; //for array utilities

public class grdtrutheval
{

    public static void main(String[] args)
    {

    //Display usage information if no argument is used
    if(args.length == 0)
    {
    System.out.print('\n');
    System.out.println("Usage: java grdtrutheval [number of reference entry]");
    System.out.println("The arguments must contain an integer from 1 to number of entries in the
file.");
    System.out.println("Output suitable for redirection to a spreadsheet (tab separted values).");
    System.out.println("E.g. java grdtrutheval 1 > data.xls");
    System.out.print('\n');
    System.exit(1);
    }
        int numberOfImages = 1; //Read from desc. file
        //Declare the results array
        int[] ImagePoints = new int[numberOfImages];

        String[] imageDesc = readFromFile("descriptions");//default file name
        //for(String temp:imageDesc)System.out.println(temp + ' ');//testing purposes

        String[][] image1phrases = extractPhrases(imageDesc);

        int i = 0;
        while(image1phrases[i] != null)
        {
            //for(String temp:image1phrases[i])System.out.print(temp + '/');//testing purposes
            System.out.print(calculatePoints(image1phrases[i],image1phrases[Integer.parseInt(args
[0])-1]));//take the reference file number from the argument
            System.out.print('\t');//tab separate values
            i++;
        }
        System.out.print('\n');
    }

    public static String[][] extractPhrases(String[] imagePhrases)//This method takes a string of
phrases and returns an array of them
    {
        String[][] allPhraseArray = new String[imagePhrases.length][];//Needs to be smaller, minus
superfluous lines.
        int finalcount = 0;

        for(int i = 0 ; i< imagePhrases.length-1 ; i++)//array returns 1 too big for unknown reason
        {
            //System.out.print(imagePhrases.length);
            //System.out.println(imagePhrases[i]);
            imagePhrases[i] = imagePhrases[i].substring(3);//remove 1st 3 chars i.e. image number

            if(imagePhrases[i+1].startsWith(" "))//test for a continuation of an entry
            {
                allPhraseArray[finalcount] = (imagePhrases[i]+imagePhrases[i+1]).split(",");//split
comma separted elements
                i++;//skip next line
            }
            else
                {
                allPhraseArray[finalcount] = imagePhrases[i].split(",");//split comma separted
elements
                }
            finalcount++;//increment output counter

        }
```

```
        int i=0;
        while(allPhraseArray[i] != null)
        {
            //System.out.print(i + " ");
            for(int j=0; j<allPhraseArray[i].length; j++)
            {
                allPhraseArray[i][j] = allPhraseArray[i][j].trim(); //Remove WS from the ends of
all elements
                while(allPhraseArray[i][j].contains("  "))allPhraseArray[i][j] = allPhraseArray[i]
[j].replaceAll("  "," "); //Makes sure that there is only one space between words
            }
            i++;
        }

        return allPhraseArray;
    }

    public static String[] readFromFile(String filename) //This method reads the descriptions file
and put the lines into an array of strings.
    {
        //System.out.print(filename);
        int linecount = 0;

        try
        {
            BufferedReader in = new BufferedReader(new FileReader(filename));
            while(in.readLine()!= null)//<>EOF
            {
                linecount++; //Find number of lines by steping through the file. There must be an
easier way.
            }
            in.close();
        }
        catch(IOException e)
        {
            //MAYBE I SHOULD PUT SOMETHING HERE
        }

        String[] phrasesLines = new String[linecount]; //declare and initialize the output now we
know the size
        linecount = 0; //reset count

        try
        {
            BufferedReader in = new BufferedReader(new FileReader(filename));
            String line;
            while((line = in.readLine())!= null)//<>EOF
            {
                //System.out.println(line);
                line = line.toLowerCase();//constant case to ease matching later.
                phrasesLines[linecount]=line;//DESC MAY BE ACROSS MULTIPLE LINES!
                linecount++; //Use same var for array index
            }
            in.close();
        }
        catch(IOException e)
        {
            //AND HERE
        }

        return phrasesLines;
    }

    public static int calculatePoints(String[] Phrases, String[] refPhrases)//This method assigns
points and gives out gold stars :)
    {
       int add = 2;//amount to add
       int sub = 1;//amount to subtract

        int points = 0;
        //equate phrases
        Arrays.sort(refPhrases);//sort array so search works
        Arrays.sort(Phrases);//sort other array so search works
        //for(String refPhrase:refPhrases)System.out.println(refPhrase);//test
```

```
        //System.out.println(Phrases.length);//test

        for(int i=0;i<Phrases.length;i++)
        {

            //System.out.println(Arrays.binarySearch(refPhrases,Phrases[i])); //test
//          ++ for similar
            if(Arrays.binarySearch(refPhrases,Phrases[i])>=0)points+=add; //Search and increment.
//          -- for dis-similar
            else points-=sub;
        }

        for(int i=0;i<refPhrases.length;i++)
        {
            //Other way round
            if(Arrays.binarySearch(Phrases,refPhrases[i])<0)points-=sub; //Search and decrement.
            else points+=add; //Alternative to +=(2*add) above
        }

        return points;
    }

}
```

D. Scaled matching component source code

```
/*
MPEG-7 Scalable CBIR Project: Matching component for resized and compressed images
James Singleton (C) 2007
*/

import javax.xml.xpath.*;
import javax.xml.parsers.*;
import org.w3c.dom.*;

public class WTl2norm
{
    public static void main(String[] args)
    throws ParserConfigurationException, XPathExpressionException, org.xml.sax.SAXException,
java.io.IOException
    {

    //Display usage information if no argument is used
    if(args.length != 1)
    {
    System.out.print('\n');
    System.out.println("Usage: java WTl2norm [filename prefix]");
    System.out.println("The argument must contain the start of the filenames of the mp7 xml
files.");
    System.out.println("Output suitable for redirection to a spreadsheet (tab separted values).");
    System.out.println("E.g. java WTl2norm image.jpg > image.xls");
    System.out.print('\n');
    System.exit(1);
    }

        String imname = args[0];//filename in argument

        String queryxmlfile =  imname+"_00.mp7.xml"; // Reference data file to which all others are
compared (full resolution and lossless)

        //Arrary of the strings needed to access the required data in the xml file for each
descriptor
        String[] desciptXpaths = { //Xpaths to locate the descriptor's tags in the XML files
        "//VisualDescriptor/Coeff/text()",  //SCD
        "//VisualDescriptor/YDCCoeff",      //CLD x 6 Y, Cb and Cr (AC and DC)
        "//VisualDescriptor/YACCoeff63",
        "//VisualDescriptor/CrDCCoeff",
        "//VisualDescriptor/CrACCoeff63",
        "//VisualDescriptor/CbDCCoeff",
        "//VisualDescriptor/CbACCoeff63",
        "//VisualDescriptor/BinCounts",     //EHD
        };

        //Weightings for each descriptor. Best just to use 1 or 0 to isolate descriptors here.
        double[] descWeightings ={0,1,1,1,1,1,1,0,};//CLD //{1,0,0,0,0,0,0,0};//SCD //
{0,0,0,0,0,0,0,1};//EHD

        String[] xmlfiles = { //Names of resized and compressed images to read. Generated from
original image name.
        imname+"_00.mp7.xml",   //full resolution and lossless
        imname+"_05.mp7.xml",   //medium quality    (q=5)
        imname+"_08.mp7.xml",   //low quality       (q=8)
        imname+"_10.mp7.xml",   //quarter resolution and lossless
        imname+"_15.mp7.xml",   //medium quality    (q=5)
        imname+"_18.mp7.xml",   //low quality       (q=8)
        imname+"_20.mp7.xml",   //sixteenth resolution and lossless
        imname+"_25.mp7.xml",   //medium quality    (q=5)
        imname+"_28.mp7.xml",   //low quality       (q=8)
        };

        double[][] L2s = new double[xmlfiles.length][desciptXpaths.length + 1]; //2D array to store
results

        for(int count2 = 0 ; count2 < desciptXpaths.length ; count2++) //Iterate through the
descriptors
        {
            String tagstring = GetXmlFiles(queryxmlfile,desciptXpaths[count2]); //get the data for
a descriptor from the reference file
            //System.out.println(tagstring);

            for(int count = 0 ; count < xmlfiles.length ; count++) //Iterate through the files
```

```
            {
                String tagstring2 = GetXmlFiles(xmlfiles[count],desciptXpaths[count2]); //get the
data for a descriptor from a file

                //System.out.println(tagstring2);
                //for(double tmp:StringtoArray(tagstring2))System.out.print(tmp + "\t");
                //System.out.print('\n');

                //Compute the L2Norm of the same descriptor for 2 files (reference and sample) and
store in 2D array
                L2s[count][count2] = L2NormArrays(StringtoArray(tagstring),StringtoArray
(tagstring2));
                //System.out.println(L2s[count][count2]); //Print values. Testing only
            }
        }


        //INTERMEDIATE TESTING OF VALUES
        //for(double tmp : descWeightings)System.out.print(tmp + ":");
        //System.out.print('\t');
        double TotalWeightings = 0.0; //Used to calculate the average later, keep at zero
        for(double temptot:descWeightings) TotalWeightings += temptot; //Calculate Total from
weightings array
        //System.out.println(TotalWeightings);

        for(int count = 0 ; count < xmlfiles.length ; count++) //Display the results to stdout for
each file
        {
        //Ajust Weighting here. Acc L2n with bias. (from array)
            double Sum = 0.0; // tempary variable to store the sum of L2Norms for each descriptor
            for(int count2 = 0 ; count2 < desciptXpaths.length ; count2++) //cycle through the
descriptors
            {
                //System.out.println(L2s[count][count2]);
                Sum += (L2s[count][count2])*descWeightings[count2]; //Accumulate the results.
            }
            //System.out.println(Sum);// Test intermediate value.
            L2s[count][desciptXpaths.length] = Sum/TotalWeightings; //Store average L2N in another
element in the array.

            //System.out.println(xmlfiles[count]);
            System.out.print(L2s[count][desciptXpaths.length]);//print the value
            System.out.print('\t');//tabs to separate the values
            //System.out.print('\n');
        }

    System.out.print('\n');//newline to end the entry
    }


    public static double L2NormArrays(int[] a, int[] b)
    // A neater implementation which also includes some error handleing.
    {
        if(a.length == b.length)
        {
            double l2n = 0.0;
            for(int i = 0 ; i < a.length ; i++)l2n += Math.pow(a[i]-b[i],2);//raise difference to
power (2) AND SUM, no need for ABS
            //System.out.println(l2n); // Test intermediate value.
            l2n = Math.sqrt(l2n);// works out the l2norm, could be modified to support lpnorm
(could use math.pow(x,(1/p)?)
            return l2n;
        }
        else
        {
            System.out.println("Arrays not of equal length");
            return -1.0;
            // Would it be better to pad the smaller array with zeros? Will they ever be unequal?
        }
    }

    public static int[] StringtoArray(String ints)
    {
        ints = ints.trim();//trims whitespce to prevent certain errors occuring in the next step.
        String temp[] = ints.split("\\s"); //Tokenizes (splits) the string into an array of
```

```
strings. No whitespace.
        int tmplngth = temp.length; //Allows only a subset of the data to be used, e.g. only the
1st 16 coefficients
        int[] result = new int[tmplngth]; //Init to size of split string array
        //for(String n:temp)System.out.println(Integer.parseInt(n));
        for(int i=0; i<tmplngth; i++)result[i] = Integer.parseInt(temp[i]);
        return result;
    }

    public static String GetXmlFiles(String documentName, String expression)
    throws ParserConfigurationException, XPathExpressionException, org.xml.sax.SAXException,
java.io.IOException
        {
            // Parse the document to a DOM tree
            // XPath can also be used with a SAX InputSource
            DocumentBuilder parser = DocumentBuilderFactory.newInstance().newDocumentBuilder();
            Document doc = parser.parse(new java.io.File(documentName));

            // Get an XPath object to evaluate the expression
            XPath xpath = XPathFactory.newInstance().newXPath();

            return(xpath.evaluate(expression, doc));
        }
}
```

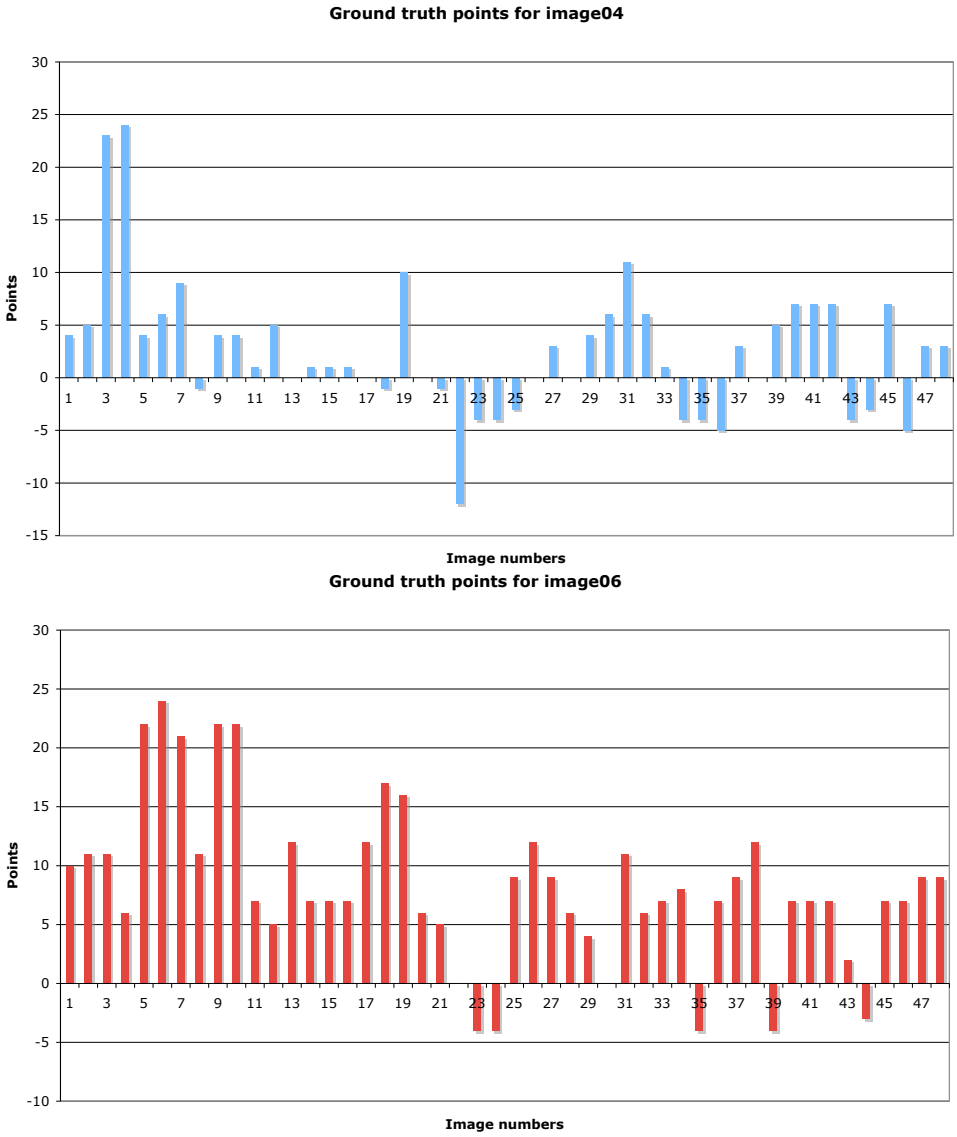
E. Average difference in L_2Norm values

columbiagorge/P6290010.jpg arborgreens/Image01.jpg barcelona/Image03.jpg campusinfall/Image06.jpg cannonbeach/Image07.jpg cherries/Image08.jpg football/Image11.jpg geneva/Image12.jpg greenlake/Image13.jpg

Res/Qual	00	05	08	10	15	18	20	25	28
Average CLD	0.00	6.64	8.08	5.54	6.17	8.63	6.70	6.56	9.11
Average SCD	0.00	7.92	12.21	9.03	9.88	15.42	13.98	13.33	18.72
Average EHD	0.00	6.62	8.34	5.70	6.41	9.32	7.17	7.37	10.28

F. Ground truth data and results

01. partly cloudy sky, building, tree, tree trunk, bush, people, grass, HUB area
02. clear sky, tree, grass, people, buildings, leafless trees, HUB area
03. clear sky, building, leafless trees, bush, sidewalk, grass, HUB area
04. clear sky, building, leafless trees, bush, sidewalk, HUB area
05. clear sky, building, tree, grass, sidewalk, people, post, quad
06. clear sky, building, tree, grass, sidewalk, people
07. clear sky, building, tree, bush, grass, sidewalk, people, post, quad
08. clear sky, building tree, leafless tree, grass, sidewalk, people, quad
09. clear sky, building, tree, leafless tree, grass, sidewalk, people, quad
10. clear sky, building, tree, leafless tree, grass, sidewalk, people, quad
11. clear sky, building, ground, people, Red Square
12. clear sky, building, ground, statue, people, bush, grass truck
13. clear sky, building, ground, tree, people, Red Square
14. clear sky, building, ground, people, Red Square
15. clear sky, building, ground, people, Red Square
16. clear sky, building, ground, people, Red Square
17. clear sky, building, ground, people, tree, Red Square
18. partly cloudy sky, tree, leafless tree, building, sidewalk, people, grass
19. clear sky, building, tree, leafless tree, bush, sidewalk, grass, tree trunk
20. clear sky, tree, building, water, ducks, Duck Pond
21. clear sky, tree, building, water, ducks, fence, Duck Pond
22. partly cloudy, sky, tree, leafless tree, grass, plants
23. partly cloudy sky, tree, tree trunk, bush
24. partly cloudy sky, tree, tree trunk, bush
25. clear sky, building, tree, leafless tree, tree trunk, grass, street, red tree, Engineering
26. clear sky, building, tree trunk, tree, grass, Engineering
27. clear sky, tree, bush, red tree, street, building, car, people, Engineering
28. clear sky, red tree, tree, building, street, Engineering
29. clear sky, building, leafless tree, red tree, tree, bush, grass sidewalk, Engineering
30. clear sky, building, red tree, bush, post, Engineering
31. clear sky, building, leafless tree, tree, bush, sidewalk, Engineering
32. clear sky, building, leafless tree, street, sidewalk, HUB
33. clear sky, building, people, atruim, HUB
34. tree, tree trunk street, building, people
35. partly cloudy sky, tree trunk, leafless tree, building
36. partly cloudy sky, tree, sidewalk, people, street
37. clear sky, building, grass
38. clear sky, building red tree, leafless tree, grass, people, sidewalk
39. clear sky, bush, tree leafless tree, bush
40. clear sky, building, tree, bush, Sieg
41. clear sky, building, tree, bush, Sieg
42. clear sky, building, tree, bush, Sieg
43. clear sky, tree, wall, street
44. building, bicycles, Sieg
45. clear sky, building, tree, bush, Sieg
46. clear sky, tree, red tree, people, wall
47. clear sky, tree, building
48. clear sky, tree, building



Graphs generated from the points for two images after processing the ground truth descriptions from the image database in the campusinfall image set.

G. JPEG2000 emulator wrapper source code

```
function imoutrgb = WTit(imagename,res,qual)
%A script to perform JPEG2000 emulation on an image.
%Usage: WTit(filename of original image, resolution setting, quality setting)
%E.g. imwrite(WTit(imagefilename,2,8),[imagefilename,'_28.jpg'],'png');
%James Singleton (C)2007

image = imread(imagename);%load image in
%imshow(image);
%imagedbl= double(image/255);%double causes error in compress
%imshow(imagedbl);
resized = imresize(image,[512 768]);

%PAD
%resized = horzcat(image,zeros(504,12,3));
%resized = vertcat(resized,zeros(8,768,3));

%size(image)
%size(resized)
imycbcr = rgb2ycbcr(resized);%to YUV
%imshow(imycbcr);
%compress x3, 5 levels
[wty,cry] = compress(imycbcr(:,:,1),5);
[wtcb,crcb] = compress(imycbcr(:,:,2),5);
[wtcr,crcr] = compress(imycbcr(:,:,3),5);
%extract x3, 5 levels
[imY,cr] = extractdecode(wty,5,res,qual);
[imCb,cr] = extractdecode(wtcb,5,res,qual);
[imCr,cr] = extractdecode(wtcr,5,res,qual);
%build output image
imout = ones(512/((2^res)),768/((2^res)),3);
imout(:,:,1) = imY;
imout(:,:,2) = imCb;
imout(:,:,3) = imCr;
%imshow(imout/255);
imoutrgb = ycbcr2rgb(imout/255);%to RGB and normalize
%imshow(imoutrgb);
```

H. Image display component source code

```
function imagedisplay(filenum)
%Displays the top images and generates a table of values.
%Usage: The parameter is the number of the image in the set to be tested.
%James Singleton (C)2007

%filename = ['Image',num2str(filenum,'%05.2d'),'.mp7.xml'];%Non scalable
filename = ['imagetest',num2str(filenum),'_00.mp7.xml'];%set res & qual of image

hoz = 5;
vert = 1;

topx = hoz*vert;
%Get retrieved results
[status,result2] = dos(['java l2norm ', filename]);
A = str2num(result2);
[SA,IX] = sort(A);
%Make a latex table
matrix2latex([SA(1,1:topx);IX(1,1:topx)],[filename,'COMBI.tex'],'rowLabels',{'L2Norms','Image numbers'})
counter = 1;
for m = 1:topx
    imnum = num2str(IX(m),'%05.2d');
    filename = ['Image',imnum,'.jpg'];
    %show images
    subplot(vert,hoz,counter); imshow(filename);
    counter = counter + 1;
end
```

I. *Performance evaluation metrics source code*

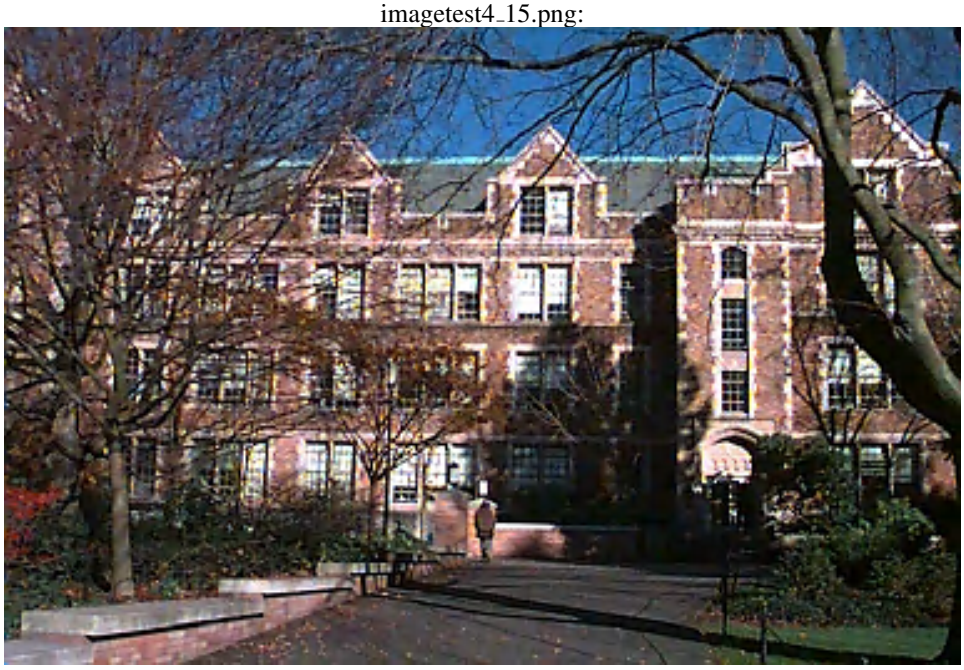
```
function metrics(filenum)
%This ties together the java modules and works out precision and recall.
%Usage: The parameter is the number of the image in the set to be tested.
%James Singleton (C)2007

%filename = ['Image',num2str(filenum,'%05.2d'),'.mp7.xml '];%For non-scalable
filename = ['imagetest',num2str(filenum),'_15.mp7.xml'];%Set res & qual here
[status,result2] = dos(['java l2norm ', filename]);%Get retrieved results
A = str2num(result2)
Range = max(A)-min(A)
[status,resultg] = dos(['java grdtrutheval ', num2str(filenum)]);%Get relevant results
B = str2num(resultg);
RangeB = max(B)-min(B);
[SA,IX] = sort(A);
hoz = 4;
vert = 3;
counter = 1;
for relthresh = 2:5
    relB = B>=(min(B)+(RangeB/relthresh));
    rel = sum(relB);
    count = 1;
    for retthresh = 1:0.1:20
        retA = A<=(max(A)-(Range/retthresh));
        ret = sum(retA);
        if (ret < 1)
            continue
        end
        %number of retrieved images that are also relevant
        retrel = sum(bitand(retA,relB));
        recall = retrel/rel;
        precision = retrel/ret;
        PR(1,count) = recall;
        PR(2,count) = precision;
        count = count+1;
        if (recall == 1)
            break
        end
    end
    subplot(vert,hoz,counter); plot(PR(1,:),PR(2,:));%plot precision vs recall
    topx = (vert*hoz)-counter;
    counter = counter + 1;
end
for m = 1:topx
    imnum = num2str(IX(m),'%05.2d');
    filename = ['Image',imnum,'.jpg'];%build image file name from index number
    %show image
    subplot(vert,hoz,counter); imshow(filename);
    counter = counter + 1;
end
```

J. *Scaled images*

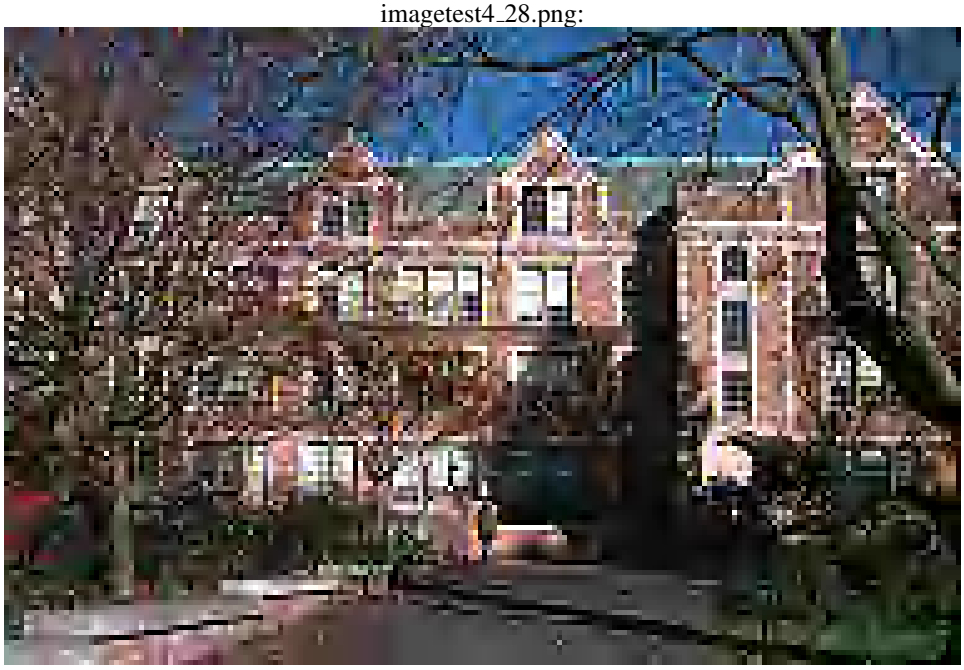
The following images have been scaled to fit on the paper although they have different resolutions.

imagetest4_00.png:

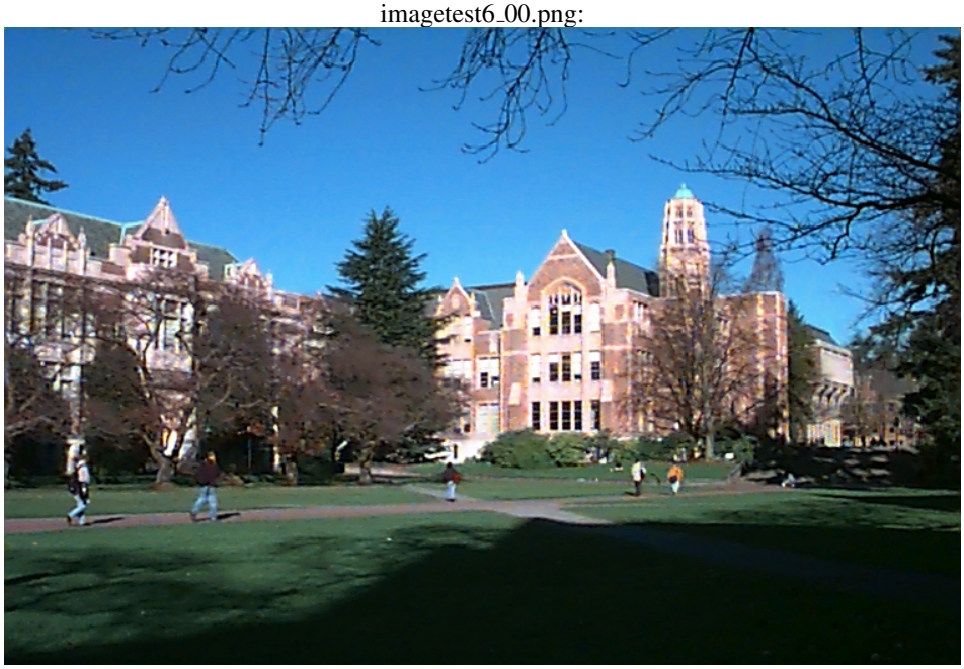
imagetest4_15.png:



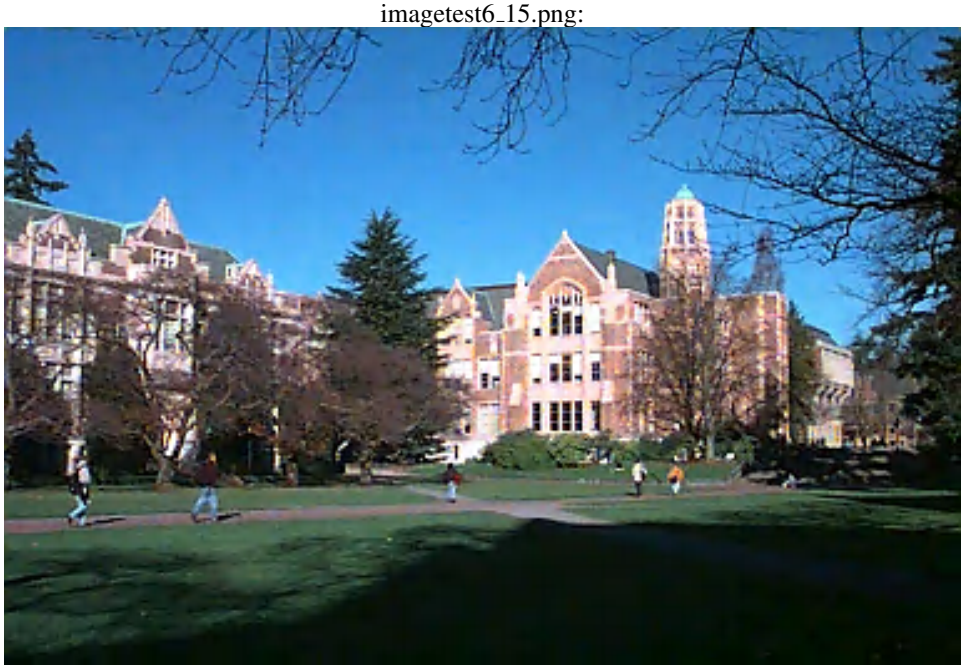
imagetest4_28.png:



imagetest6_00.png:



imagetest6_15.png:



imagetest6_28.png:

K. MPEG-7 XML files

```
<?xml version="1.0" encoding="UTF-8"?>
<Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2001" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <DescriptionMetadata>
    <Version>2.0</Version>
    <Instrument>
      <Tool>
        <Name>Caliph v0.9.23</Name>
      </Tool>
    </Instrument>
  </DescriptionMetadata>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="ImageType">
      <Image>
        <MediaInformation>
          <MediaProfile master="true">
            <MediaFormat>
              <Content href="image" />
              <FileFormat href="urn:mpeg:MPEG7FileFormatCS:1">
                <Name />
              </FileFormat>
              <VisualCoding>
                <Format href="urn:mpeg:MPEG7FileFormatCS:1" colorDomain="color">
                  <Name />
                </Format>
                <Pixel bitsPer="24" />
                <Frame width="768" height="512" />
              </VisualCoding>
            </MediaFormat>
            <MediaInstance>
              <InstanceIdentifier />
              <MediaLocator>
                <MediaUri>file:/C:/Documents%20and%20Settings/James/Desktop/BEng%20Project/
imagetest4_00.png</MediaUri>
              </MediaLocator>
            </MediaInstance>
          </MediaProfile>
          <MediaProfile>
            <MediaFormat>
              <Content href="image" />
              <FileFormat href="urn:mpeg:MPEG7FileFormatCS:1">
                <Name>JPEG</Name>
              </FileFormat>
              <VisualCoding>
                <Format href="urn:mpeg:MPEG7FileFormatCS:1" colorDomain="color">
                  <Name>JPEG</Name>
                </Format>
                <Pixel bitsPer="24" />
                <Frame width="120" height="120" />
              </VisualCoding>
            </MediaFormat>
            <MediaInstance>
              <InstanceIdentifier />
              <MediaLocator>
                <MediaUri>file:/C:/Documents%20and%20Settings/James/Desktop/BEng%20Project/
tn_imagetest4_00.png</MediaUri>
              </MediaLocator>
            </MediaInstance>
          </MediaProfile>
        </MediaInformation>
        <VisualDescriptor xsi:type="ColorLayoutType">
          <YDCCoeff>13</YDCCoeff>
          <CbDCCoeff>30</CbDCCoeff>
          <CrDCCoeff>33</CrDCCoeff>
          <YACCoeff63>16 20 12 11 11 14 14 18 11 16 15 18 17 15 17 16 16 17 17 14 17 15 17 13 17 15
14 16 15 14 15 17 13 17 14 15 15 16 15 17 17 16 15 16 15 17 15 16 17 14 15 15 15 16 15 16 15 16 16
16 15 15 16</YACCoeff63>
          <CbACCoeff63>15 20 20 14 14 16 14 15 18 16 15 14 16 16 15 16 16 15 15 15 15 16 16 16 16
15 16 15 16 15 16 16 16 16 15 16 16 15 16 15 16 15 15 16 15 16 15 16 16 16 15 16 16 16 16 15 16 15
16 16 16 16 16</CbACCoeff63>
          <CrACCoeff63>18 13 11 16 16 15 18 17 13 15 16 17 15 15 16 15 15 16 16 16 17 15 15 15 16
16 15 16 15 15 16 16 15 16 16 15 15 16 15 16 16 16 15 16 16 16 16 15 16 15 16 15 15 15 15 16 16 16
16 15 15 15 15</CrACCoeff63>
        </VisualDescriptor>
        <VisualDescriptor xsi:type="ScalableColorType" numOfBitplanesDiscarded="0"
numOfCoeff="256">
```

```
          <Coeff>-202 71 27 54 1 7 22 29 13 14 9 21 9 14 19 22 -7 -8 -3 -1 -7 5 -4 -1 -15 -10 -4 0
 -15 0 1 -4 0 -3 0 2 3 1 5 3 -1 2 -7 3 -8 -1 3 4 1 -3 -1 -6 -5 3 -1 0 -2 0 -10 -2 -12 0 -6 -3 0 1 2
 5 1 1 0 1 1 1 2 0 2 0 3 2 0 -1 0 -1 -1 -1 2 1 15 6 9 5 7 3 3 0 0 -1 0 0 3 -3 2 1 2 -1 2 -5 1 3 -1 0
 -1 0 -3 1 -3 0 0 0 1 1 1 0 1 0 -1 1 0 -1 -2 -3 -1 0 0 0 0 -1 0 -6 -1 -1 -1 0 0 -1 0 -6 -1 -3 0 -1 0
 -3 0 -6 -2 -3 0 1 0 -1 0 -3 -1 -1 -3 1 0 -1 0 -4 -1 2 -2 0 -3 1 -1 2 -3 0 -2 -2 -2 -3 -3 0 0 -3 0
 -1 0 -1 0 -3 -1 -3 1 1 0 -1 0 -7 -2 -3 -1 -1 0 -1 1 -10 0 -1 -1 -2 0 -3 0 -7 -2 1 -2 -1 -1 -1 1 -3
 -2 0 -2 -1 0 -1 1 -7 3 0 0 -1 -2 -1 -3 0 0 1 0 -1 -2 -3 -3 0 0 1 0 1</Coeff>
        </VisualDescriptor>
        <VisualDescriptor xsi:type="EdgeHistogramType">
          <BinCounts>3 2 6 5 7 3 3 5 4 7 2 4 5 3 7 5 2 2 6 7 3 1 6 6 6 3 4 5 7 4 4 4 6 6 5 6 2 4 5
 6 3 2 6 7 6 3 4 6 5 6 5 3 6 5 5 3 2 6 6 7 2 6 6 4 4 1 5 4 6 6 2 6 3 6 4 1 6 5 4 5</BinCounts>
        </VisualDescriptor>
        <VisualDescriptor xsi:type="DominantColorType">
          <SpatialCoherency>0</SpatialCoherency>
          <Value>
            <Percentage>3</Percentage>
            <Index>7 12 17</Index>
          </Value>
          <Value>
            <Percentage>8</Percentage>
            <Index>8 7 7</Index>
          </Value>
          <Value>
            <Percentage>7</Percentage>
            <Index>2 2 1</Index>
          </Value>
          <Value>
            <Percentage>2</Percentage>
            <Index>17 16 16</Index>
          </Value>
          <Value>
            <Percentage>2</Percentage>
            <Index>21 16 15</Index>
          </Value>
          <Value>
            <Percentage>2</Percentage>
            <Index>29 27 26</Index>
          </Value>
          <Value>
            <Percentage>4</Percentage>
            <Index>12 9 8</Index>
          </Value>
        </VisualDescriptor>
      </Image>
    </MultimediaContent>
  </Description>
</Mpeg7>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2001" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <DescriptionMetadata>
    <Version>2.0</Version>
    <Instrument>
      <Tool>
        <Name>Caliph v0.9.23</Name>
      </Tool>
    </Instrument>
  </DescriptionMetadata>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="ImageType">
      <Image>
        <MediaInformation>
          <MediaProfile master="true">
            <MediaFormat>
              <Content href="image" />
              <FileFormat href="urn:mpeg:MPEG7FileFormatCS:1">
                <Name />
              </FileFormat>
              <VisualCoding>
                <Format href="urn:mpeg:MPEG7FileFormatCS:1" colorDomain="color">
                  <Name />
                </Format>
                <Pixel bitsPer="24" />
                <Frame width="384" height="256" />
              </VisualCoding>
            </MediaFormat>
            <MediaInstance>
              <InstanceIdentifier />
              <MediaLocator>
                <MediaUri>file:/C:/Documents%20and%20Settings/James/Desktop/BEng%20Project/
imagetest4_15.png</MediaUri>
              </MediaLocator>
            </MediaInstance>
          </MediaProfile>
          <MediaProfile>
            <MediaFormat>
              <Content href="image" />
              <FileFormat href="urn:mpeg:MPEG7FileFormatCS:1">
                <Name>JPEG</Name>
              </FileFormat>
              <VisualCoding>
                <Format href="urn:mpeg:MPEG7FileFormatCS:1" colorDomain="color">
                  <Name>JPEG</Name>
                </Format>
                <Pixel bitsPer="24" />
                <Frame width="120" height="120" />
              </VisualCoding>
            </MediaFormat>
            <MediaInstance>
              <InstanceIdentifier />
              <MediaLocator>
                <MediaUri>file:/C:/Documents%20and%20Settings/James/Desktop/BEng%20Project/
tn_imagetest4_15.png</MediaUri>
              </MediaLocator>
            </MediaInstance>
          </MediaProfile>
        </MediaInformation>
        <VisualDescriptor xsi:type="ColorLayoutType">
          <YDCCoeff>13</YDCCoeff>
          <CbDCCoeff>30</CbDCCoeff>
          <CrDCCoeff>34</CrDCCoeff>
          <YACCoeff63>16 20 12 11 11 14 14 18 11 16 15 18 17 15 17 16 16 17 17 14 17 15 17 13 17 15
14 16 15 14 15 17 13 17 14 15 15 16 15 17 17 16 15 16 15 17 15 16 17 14 15 15 15 16 15 16 15 16 16
15 15 15 16</YACCoeff63>
          <CbACCoeff63>15 20 20 15 14 16 14 15 18 16 15 14 16 16 15 16 16 15 16 15 15 16 16 16 16
15 16 15 16 15 16 16 16 16 15 16 16 15 16 15 16 15 15 16 15 16 15 16 16 16 15 16 16 16 16 15 16 15
16 16 16 16 16</CbACCoeff63>
          <CrACCoeff63>18 13 11 16 16 15 18 17 13 15 16 17 15 15 16 15 15 16 16 16 17 15 15 15 16
16 15 16 15 15 16 16 14 16 16 15 15 16 15 16 16 16 15 16 16 16 16 15 16 15 16 15 15 15 15 16 16 16
16 15 15 15 15</CrACCoeff63>
        </VisualDescriptor>
        <VisualDescriptor xsi:type="ScalableColorType" numOfBitplanesDiscarded="0"
numOfCoeff="256">
```

```
          <Coeff>-202 71 27 54 -1 5 17 25 12 14 11 22 8 13 19 22 -7 -4 -3 0 -4 7 -2 -2 -15 -10 -6
 -4 -15 -3 1 -4 0 -3 -2 2 0 1 4 4 -2 1 -7 3 -5 1 5 5 1 -3 0 -6 -3 -2 -1 -4 -2 -4 -7 -6 -12 0 -10 -4
 0 1 2 3 1 1 3 1 1 1 2 0 2 1 1 1 0 -1 -2 -1 -1 -1 2 0 13 4 9 4 10 4 3 -1 0 -2 0 0 1 -1 2 0 2 1 2 -3
 1 -1 0 -2 -1 0 -2 1 -3 0 -1 -1 1 1 1 0 1 0 -1 1 0 -1 -2 -3 -1 0 0 2 0 -1 0 -4 -1 0 -2 2 0 -1 0 -4
 -1 -2 2 2 0 -3 0 -5 -1 -2 -1 4 0 -1 0 -3 -1 0 -3 3 0 -1 -2 -2 -1 3 -2 3 -3 1 -3 5 -3 3 -1 2 -3 -2
 -3 1 -1 -1 2 0 0 -1 0 -3 0 -3 -1 1 0 -1 0 -7 -2 -3 0 -2 0 -1 1 -8 0 -1 -1 -3 0 -3 0 -7 -2 -2 -2 -1
 -1 -1 1 -3 -1 -3 -1 0 0 -1 2 -7 2 -2 4 -1 -2 -1 -2 -1 -2 1 0 -1 -2 -3 -3 0 0 1 0 1</Coeff>
        </VisualDescriptor>
        <VisualDescriptor xsi:type="EdgeHistogramType">
          <BinCounts>4 2 5 6 7 2 4 6 3 7 2 5 4 4 7 4 3 3 7 6 3 1 6 4 7 2 4 7 6 5 6 3 5 4 4 5 2 3 6
 6 3 1 6 7 7 4 2 6 5 7 6 2 6 4 6 2 3 6 7 6 3 5 5 2 6 1 5 3 4 7 2 6 4 5 5 3 4 4 6 6</BinCounts>
        </VisualDescriptor>
        <VisualDescriptor xsi:type="DominantColorType">
          <SpatialCoherency>0</SpatialCoherency>
          <Value>
            <Percentage>2</Percentage>
            <Index>7 12 18</Index>
          </Value>
          <Value>
            <Percentage>3</Percentage>
            <Index>8 8 10</Index>
          </Value>
          <Value>
            <Percentage>7</Percentage>
            <Index>2 2 1</Index>
          </Value>
          <Value>
            <Percentage>1</Percentage>
            <Index>21 21 21</Index>
          </Value>
          <Value>
            <Percentage>2</Percentage>
            <Index>21 16 14</Index>
          </Value>
          <Value>
            <Percentage>7</Percentage>
            <Index>8 7 6</Index>
          </Value>
          <Value>
            <Percentage>4</Percentage>
            <Index>14 12 12</Index>
          </Value>
          <Value>
            <Percentage>1</Percentage>
            <Index>30 28 26</Index>
          </Value>
        </VisualDescriptor>
      </Image>
    </MultimediaContent>
  </Description>
</Mpeg7>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2001" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <DescriptionMetadata>
    <Version>2.0</Version>
    <Instrument>
      <Tool>
        <Name>Caliph v0.9.23</Name>
      </Tool>
    </Instrument>
  </DescriptionMetadata>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="ImageType">
      <Image>
        <MediaInformation>
          <MediaProfile master="true">
            <MediaFormat>
              <Content href="image" />
              <FileFormat href="urn:mpeg:MPEG7FileFormatCS:1">
                <Name />
              </FileFormat>
              <VisualCoding>
                <Format href="urn:mpeg:MPEG7FileFormatCS:1" colorDomain="color">
                  <Name />
                </Format>
                <Pixel bitsPer="24" />
                <Frame width="192" height="128" />
              </VisualCoding>
            </MediaFormat>
            <MediaInstance>
              <InstanceIdentifier />
              <MediaLocator>
                <MediaUri>file:/C:/Documents%20and%20Settings/James/Desktop/BEng%20Project/
imagetest4_28.png</MediaUri>
              </MediaLocator>
            </MediaInstance>
          </MediaProfile>
          <MediaProfile>
            <MediaFormat>
              <Content href="image" />
              <FileFormat href="urn:mpeg:MPEG7FileFormatCS:1">
                <Name>JPEG</Name>
              </FileFormat>
              <VisualCoding>
                <Format href="urn:mpeg:MPEG7FileFormatCS:1" colorDomain="color">
                  <Name>JPEG</Name>
                </Format>
                <Pixel bitsPer="24" />
                <Frame width="120" height="120" />
              </VisualCoding>
            </MediaFormat>
            <MediaInstance>
              <InstanceIdentifier />
              <MediaLocator>
                <MediaUri>file:/C:/Documents%20and%20Settings/James/Desktop/BEng%20Project/
tn_imagetest4_28.png</MediaUri>
              </MediaLocator>
            </MediaInstance>
          </MediaProfile>
        </MediaInformation>
        <VisualDescriptor xsi:type="ColorLayoutType">
          <YDCCoeff>13</YDCCoeff>
          <CbDCCoeff>30</CbDCCoeff>
          <CrDCCoeff>33</CrDCCoeff>
          <YACCoeff63>16 20 12 11 11 14 15 18 11 16 15 18 17 15 17 16 16 17 17 14 17 15 17 13 17 15
14 16 15 14 15 17 14 17 14 15 15 16 15 17 16 16 15 16 15 17 15 16 17 14 15 15 15 16 15 16 15 15 16
15 15 14 16</YACCoeff63>
          <CbACCoeff63>14 20 20 14 14 16 14 15 18 16 15 14 15 16 15 16 16 15 15 15 15 16 16 16 16
15 16 15 16 15 16 16 17 16 15 16 16 15 16 15 15 15 15 16 15 16 15 16 16 16 15 16 16 16 16 15 16 15
16 16 16 16 16</CbACCoeff63>
          <CrACCoeff63>18 13 11 16 16 15 18 17 14 15 16 17 15 15 16 16 15 16 16 16 17 15 15 15 16
16 15 16 15 15 16 15 15 16 16 15 15 16 15 16 16 16 15 16 16 16 15 15 16 15 16 15 15 15 15 16 16 16
16 15 15 15 15</CrACCoeff63>
        </VisualDescriptor>
        <VisualDescriptor xsi:type="ScalableColorType" numOfBitplanesDiscarded="0"
numOfCoeff="256">
```

```
          <Coeff>-202 66 30 51 -1 8 15 18 8 11 10 21 3 11 18 15 -5 2 1 3 -3 6 0 1 -9 -4 -6 -1 -15
 -2 3 -4 -1 -3 -3 2 -3 0 3 7 -1 -1 -4 0 -3 0 3 6 1 -3 1 -2 -1 -3 0 -2 0 -7 -3 -7 -11 -4 -14 -13 1 1
 2 2 1 1 3 1 1 0 2 2 2 2 2 0 -1 -1 -3 0 -1 -1 1 2 7 2 12 1 7 3 2 0 0 0 0 2 2 1 3 1 2 3 2 0 3 0 0 0
 -1 2 -1 3 -2 0 -1 1 1 -3 1 1 -6 1 -4 -5 -1 -1 -2 0 -1 1 -2 3 0 -1 0 -2 -1 0 0 0 0 0 0 -3 -1 -1 2 1
 0 -2 0 1 -1 1 0 2 0 -1 0 1 -1 1 -1 3 -1 -1 -2 1 -1 3 -1 3 -3 1 -3 6 -3 3 -1 1 -2 0 -3 1 -1 1 0 1 0
 -1 1 -3 -1 -3 -3 1 0 -1 0 -4 0 -3 0 -2 0 -1 1 -5 -2 0 0 -2 0 -3 0 -4 -2 -1 -1 1 -1 -1 1 -3 -1 -3 -1
 0 0 -1 1 -6 0 -3 0 1 0 -1 -1 -5 -3 3 -2 3 0 -3 -2 -1 -3 3 0 1</Coeff>
        </VisualDescriptor>
        <VisualDescriptor xsi:type="EdgeHistogramType">
          <BinCounts>3 2 5 5 7 2 3 4 6 7 1 4 5 4 7 3 3 6 5 6 3 1 5 6 7 3 5 4 5 6 6 3 5 4 5 5 2 5 5
 6 2 3 6 4 7 3 2 5 6 7 4 1 6 6 7 4 3 6 6 5 2 5 6 4 6 4 4 4 5 6 3 6 4 5 5 3 4 5 6 6</BinCounts>
        </VisualDescriptor>
        <VisualDescriptor xsi:type="DominantColorType">
          <SpatialCoherency>0</SpatialCoherency>
          <Value>
            <Percentage>1</Percentage>
            <Index>7 13 19</Index>
          </Value>
          <Value>
            <Percentage>4</Percentage>
            <Index>16 14 14</Index>
          </Value>
          <Value>
            <Percentage>7</Percentage>
            <Index>2 2 1</Index>
          </Value>
          <Value>
            <Percentage>2</Percentage>
            <Index>7 9 12</Index>
          </Value>
          <Value>
            <Percentage>10</Percentage>
            <Index>9 7 7</Index>
          </Value>
          <Value>
            <Percentage>2</Percentage>
            <Index>28 26 26</Index>
          </Value>
          <Value>
            <Percentage>2</Percentage>
            <Index>21 16 15</Index>
          </Value>
        </VisualDescriptor>
      </Image>
    </MultimediaContent>
  </Description>
</Mpeg7>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2001" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <DescriptionMetadata>
    <Version>3.0</Version>
    <Instrument>
      <Tool>
        <Name>Caliph v0.9.23</Name>
      </Tool>
    </Instrument>
  </DescriptionMetadata>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="ImageType">
      <Image>
        <MediaInformation>
          <MediaProfile master="true">
            <MediaFormat>
              <Content href="image" />
              <FileFormat href="urn:mpeg:MPEG7FileFormatCS:1">
                <Name />
              </FileFormat>
              <VisualCoding>
                <Format href="urn:mpeg:MPEG7FileFormatCS:1" colorDomain="color">
                  <Name />
                </Format>
                <Pixel bitsPer="24" />
                <Frame width="768" height="512" />
              </VisualCoding>
            </MediaFormat>
            <MediaInstance>
              <InstanceIdentifier />
              <MediaLocator>
                <MediaUri>file:/C:/Documents%20and%20Settings/James/Desktop/Tidy/BEng%20Project/
imagetest6_00.png</MediaUri>
              </MediaLocator>
            </MediaInstance>
          </MediaProfile>
          <MediaProfile>
            <MediaFormat>
              <Content href="image" />
              <FileFormat href="urn:mpeg:MPEG7FileFormatCS:1">
                <Name>JPEG</Name>
              </FileFormat>
              <VisualCoding>
                <Format href="urn:mpeg:MPEG7FileFormatCS:1" colorDomain="color">
                  <Name>JPEG</Name>
                </Format>
                <Pixel bitsPer="24" />
                <Frame width="120" height="120" />
              </VisualCoding>
            </MediaFormat>
            <MediaInstance>
              <InstanceIdentifier />
              <MediaLocator>
                <MediaUri>file:/C:/Documents%20and%20Settings/James/Desktop/Tidy/BEng%20Project/
tn_imagetest6_00.png</MediaUri>
              </MediaLocator>
            </MediaInstance>
          </MediaProfile>
        </MediaInformation>
        <VisualDescriptor xsi:type="ColorLayoutType">
          <YDCCoeff>15</YDCCoeff>
          <CbDCCoeff>43</CbDCCoeff>
          <CrDCCoeff>17</CrDCCoeff>
          <YACCoeff63>18 26 7 15 11 19 15 18 14 14 14 19 16 15 15 15 13 14 13 17 16 17 16 15 15 16
14 17 15 17 17 16 17 15 16 17 15 15 17 16 17 15 14 15 15 15 16 16 16 16 16 15 15 17 16 16 16 15 16
16 15 16 16</YACCoeff63>
          <CbACCoeff63>16 28 22 17 14 15 14 17 13 12 17 15 16 15 16 15 17 17 15 14 16 14 17 17 16
16 15 16 15 15 16 15 16 15 16 16 15 15 16 15 16 16 15 16 16 16 15 14 16 15 16 16 15 15 16 16 15 15
15 16 16 15 16</CbACCoeff63>
          <CrACCoeff63>15 5 7 15 17 17 18 14 18 21 15 16 16 16 15 16 14 14 16 16 15 17 13 14 16 15
16 15 16 16 15 16 15 16 15 15 17 17 15 16 15 15 16 15 15 15 16 16 16 16 15 15 16 16 16 16 16 16
15 15 15 15</CrACCoeff63>
        </VisualDescriptor>
        <VisualDescriptor xsi:type="ScalableColorType" numOfBitplanesDiscarded="0"
numOfCoeff="256">
```

```
          <Coeff>-202 67 28 54 8 18 22 26 8 3 2 17 5 12 19 22 -7 -4 -2 5 -7 3 -1 0 -10 -6 -7 -2 -15
 -3 1 -4 0 -3 0 1 1 3 6 6 -5 -7 -12 -7 -11 -6 3 0 1 3 1 8 -2 3 -1 -1 0 3 -3 2 -10 0 -7 -3 0 1 0 2 1
 1 0 2 1 2 2 0 2 1 2 4 0 -1 0 0 -1 0 1 1 11 4 4 -1 4 2 4 -2 0 -2 0 1 0 0 1 -1 2 2 2 1 2 3 -1 0 -1 3
 -1 3 -1 0 -1 0 2 -1 3 1 1 0 -1 1 0 -1 0 -3 -1 1 0 3 0 -1 0 -4 -1 0 0 3 0 -1 0 -5 -1 -1 3 2 0 -3 0
 -4 -2 -2 1 5 0 -1 0 -3 -1 -1 -2 3 0 -1 0 -4 -1 0 -1 2 0 0 0 1 0 0 -3 2 -3 -3 2 -4 0 0 -1 2 0 -1 0
 -3 1 -3 -1 2 0 -1 0 -7 -2 0 -2 0 0 -1 1 -15 -3 1 -3 -2 0 -3 0 -7 -3 1 -5 -1 -1 -1 1 -3 -3 0 -2 -1 0
 -1 0 -6 -1 -3 2 -1 0 -1 -1 -4 0 1 0 -1 0 -3 -1 -3 0 1 0 1</Coeff>
        </VisualDescriptor>
        <VisualDescriptor xsi:type="EdgeHistogramType">
          <BinCounts>1 7 5 0 3 2 7 3 1 4 1 6 5 5 4 2 3 5 6 7 2 4 6 6 5 3 6 4 5 5 3 5 1 6 6 3 3 4 5
 7 4 3 5 7 5 3 5 6 3 6 2 5 6 4 5 4 4 6 4 5 0 7 2 2 4 0 6 6 4 5 1 7 4 3 5 1 5 5 5 6</BinCounts>
        </VisualDescriptor>
        <VisualDescriptor xsi:type="DominantColorType">
          <SpatialCoherency>0</SpatialCoherency>
          <Value>
            <Percentage>8</Percentage>
            <Index>8 18 27</Index>
          </Value>
          <Value>
            <Percentage>8</Percentage>
            <Index>1 2 1</Index>
          </Value>
          <Value>
            <Percentage>3</Percentage>
            <Index>9 11 9</Index>
          </Value>
          <Value>
            <Percentage>1</Percentage>
            <Index>26 25 25</Index>
          </Value>
          <Value>
            <Percentage>2</Percentage>
            <Index>6 12 18</Index>
          </Value>
          <Value>
            <Percentage>4</Percentage>
            <Index>11 10 9</Index>
          </Value>
          <Value>
            <Percentage>1</Percentage>
            <Index>25 20 18</Index>
          </Value>
        </VisualDescriptor>
      </Image>
    </MultimediaContent>
  </Description>
</Mpeg7>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2001" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <DescriptionMetadata>
    <Version>3.0</Version>
    <Instrument>
      <Tool>
        <Name>Caliph v0.9.23</Name>
      </Tool>
    </Instrument>
  </DescriptionMetadata>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="ImageType">
      <Image>
        <MediaInformation>
          <MediaProfile master="true">
            <MediaFormat>
              <Content href="image" />
              <FileFormat href="urn:mpeg:MPEG7FileFormatCS:1">
                <Name />
              </FileFormat>
              <VisualCoding>
                <Format href="urn:mpeg:MPEG7FileFormatCS:1" colorDomain="color">
                  <Name />
                </Format>
                <Pixel bitsPer="24" />
                <Frame width="384" height="256" />
              </VisualCoding>
            </MediaFormat>
            <MediaInstance>
              <InstanceIdentifier />
              <MediaLocator>
                <MediaUri>file:/C:/Documents%20and%20Settings/James/Desktop/Tidy/BEng%20Project/
imagetest6_15.png</MediaUri>
              </MediaLocator>
            </MediaInstance>
          </MediaProfile>
          <MediaProfile>
            <MediaFormat>
              <Content href="image" />
              <FileFormat href="urn:mpeg:MPEG7FileFormatCS:1">
                <Name>JPEG</Name>
              </FileFormat>
              <VisualCoding>
                <Format href="urn:mpeg:MPEG7FileFormatCS:1" colorDomain="color">
                  <Name>JPEG</Name>
                </Format>
                <Pixel bitsPer="24" />
                <Frame width="120" height="120" />
              </VisualCoding>
            </MediaFormat>
            <MediaInstance>
              <InstanceIdentifier />
              <MediaLocator>
                <MediaUri>file:/C:/Documents%20and%20Settings/James/Desktop/Tidy/BEng%20Project/
tn_imagetest6_15.png</MediaUri>
              </MediaLocator>
            </MediaInstance>
          </MediaProfile>
        </MediaInformation>
        <VisualDescriptor xsi:type="ColorLayoutType">
          <YDCCoeff>15</YDCCoeff>
          <CbDCCoeff>44</CbDCCoeff>
          <CrDCCoeff>17</CrDCCoeff>
          <YACCoeff63>18 26 7 15 11 19 15 18 14 14 14 19 16 15 15 15 13 14 13 17 16 17 16 15 15 16
14 17 15 17 17 16 17 15 16 17 15 15 17 16 17 15 14 15 15 15 16 16 16 16 16 15 15 17 16 16 16 15 16
16 15 16 16</YACCoeff63>
          <CbACCoeff63>16 28 22 17 14 15 14 17 13 12 17 15 16 15 16 15 17 17 15 15 16 14 17 17 16
16 15 16 15 15 16 15 16 15 16 16 15 15 16 15 16 15 16 16 15 16 15 16 15 15 16 15 16 16 14 15 15 16 16 15
15 16 16 16 16</CbACCoeff63>
          <CrACCoeff63>15 5 7 15 17 17 18 14 18 21 15 16 16 16 15 16 13 14 16 16 15 17 13 14 16 15
16 15 16 16 15 16 15 16 15 15 17 17 15 16 15 15 16 15 15 15 16 16 16 16 15 16 16 16 16 16 16 16 16
15 15 15 15</CrACCoeff63>
        </VisualDescriptor>
        <VisualDescriptor xsi:type="ScalableColorType" numOfBitplanesDiscarded="0"
numOfCoeff="256">
```

```
          <Coeff>-202 64 28 52 8 21 21 23 7 2 5 19 6 12 16 16 -7 -1 -2 3 -4 0 -3 -1 -8 -3 -4 -1 -12
0 3 -4 -1 -3 -2 1 1 1 5 4 -5 -4 -9 -8 -6 -4 4 2 1 3 1 11 0 8 1 0 0 6 -3 3 -11 0 -11 -8 0 1 1 1 1 1
3 2 1 0 2 0 2 1 2 2 -1 -1 -2 1 -1 -1 2 1 11 2 7 -3 7 1 3 -2 0 -3 0 1 0 0 2 1 2 2 2 3 2 5 0 0 -1 1
-1 2 -1 1 0 0 2 -1 1 0 -4 0 -4 -2 0 -1 -1 -2 -1 1 0 3 0 -1 0 -2 -1 0 1 3 0 -1 0 -3 -1 -1 3 1 0 -3 0
-4 -1 -1 0 3 0 -1 0 -2 -1 -1 -1 1 -1 -1 -2 -2 -1 -1 -1 1 0 1 0 1 -2 0 -1 1 -3 -2 3 -7 1 -3 1 1 0 -1
0 -3 1 -3 -2 2 0 -1 0 -7 -2 0 -1 4 0 -1 1 -15 -2 1 -1 3 0 -3 0 -7 -3 2 -2 4 -1 -1 1 -3 -3 -1 -1 3 0
-1 0 -3 0 -1 4 3 0 -1 0 -2 0 3 3 2 1 -3 -1 -2 -1 3 0 1</Coeff>
        </VisualDescriptor>
        <VisualDescriptor xsi:type="EdgeHistogramType">
          <BinCounts>0 6 6 5 4 1 6 6 3 5 2 5 5 4 6 3 4 3 4 7 3 4 6 5 6 1 6 6 6 4 3 5 4 4 5 3 4 6 6
5 4 4 4 6 6 3 4 4 5 7 3 4 5 4 6 4 4 5 4 6 2 7 2 3 5 3 6 6 2 3 5 5 5 4 4 2 5 4 6 5</BinCounts>
        </VisualDescriptor>
        <VisualDescriptor xsi:type="DominantColorType">
          <SpatialCoherency>0</SpatialCoherency>
          <Value>
            <Percentage>8</Percentage>
            <Index>8 18 27</Index>
          </Value>
          <Value>
            <Percentage>8</Percentage>
            <Index>1 2 1</Index>
          </Value>
          <Value>
            <Percentage>3</Percentage>
            <Index>9 11 9</Index>
          </Value>
          <Value>
            <Percentage>1</Percentage>
            <Index>27 26 26</Index>
          </Value>
          <Value>
            <Percentage>2</Percentage>
            <Index>6 12 18</Index>
          </Value>
          <Value>
            <Percentage>4</Percentage>
            <Index>11 10 9</Index>
          </Value>
          <Value>
            <Percentage>1</Percentage>
            <Index>24 20 18</Index>
          </Value>
        </VisualDescriptor>
      </Image>
    </MultimediaContent>
  </Description>
</Mpeg7>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2001" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <DescriptionMetadata>
    <Version>3.0</Version>
    <Instrument>
      <Tool>
        <Name>Caliph v0.9.23</Name>
      </Tool>
    </Instrument>
  </DescriptionMetadata>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="ImageType">
      <Image>
        <MediaInformation>
          <MediaProfile master="true">
            <MediaFormat>
              <Content href="image" />
              <FileFormat href="urn:mpeg:MPEG7FileFormatCS:1">
                <Name>JPEG</Name>
              </FileFormat>
              <VisualCoding>
                <Format href="urn:mpeg:MPEG7FileFormatCS:1" colorDomain="color">
                  <Name>JPEG</Name>
                </Format>
                <Pixel bitsPer="24" />
                <Frame width="192" height="128" />
              </VisualCoding>
            </MediaFormat>
            <MediaInstance>
              <InstanceIdentifier />
              <MediaLocator>
                <MediaUri>file:/C:/Documents%20and%20Settings/James/Desktop/Tidy/BEng%20Project/
imagetest6_28.jpg</MediaUri>
              </MediaLocator>
            </MediaInstance>
          </MediaProfile>
          <MediaProfile>
            <MediaFormat>
              <Content href="image" />
              <FileFormat href="urn:mpeg:MPEG7FileFormatCS:1">
                <Name>JPEG</Name>
              </FileFormat>
              <VisualCoding>
                <Format href="urn:mpeg:MPEG7FileFormatCS:1" colorDomain="color">
                  <Name>JPEG</Name>
                </Format>
                <Pixel bitsPer="24" />
                <Frame width="120" height="120" />
              </VisualCoding>
            </MediaFormat>
            <MediaInstance>
              <InstanceIdentifier />
              <MediaLocator>
                <MediaUri>file:/C:/Documents%20and%20Settings/James/Desktop/Tidy/BEng%20Project/
tn_imagetest6_28.jpg</MediaUri>
              </MediaLocator>
            </MediaInstance>
          </MediaProfile>
        </MediaInformation>
        <VisualDescriptor xsi:type="ColorLayoutType">
          <YDCCoeff>15</YDCCoeff>
          <CbDCCoeff>44</CbDCCoeff>
          <CrDCCoeff>17</CrDCCoeff>
          <YACCoeff63>18 26 7 15 11 18 15 18 14 14 14 18 16 15 15 15 13 14 13 17 15 17 16 15 15 16
15 17 15 17 17 16 17 15 16 17 15 15 17 16 17 15 14 15 15 15 16 16 16 15 16 15 15 17 16 16 16 15 16
16 15 16 16</YACCoeff63>
          <CbACCoeff63>16 28 22 17 14 15 14 17 13 12 16 15 15 15 16 15 17 17 15 14 16 14 17 17 15
16 15 16 15 15 17 15 16 15 16 16 15 15 16 15 16 16 15 16 16 16 15 14 16 15 16 16 15 15 15 16 15 15
15 16 16 15 16</CbACCoeff63>
          <CrACCoeff63>16 5 7 15 17 17 18 14 19 20 15 17 16 16 15 16 14 14 16 16 15 17 13 14 16 15
16 15 16 16 15 16 15 16 15 15 17 16 15 16 15 15 16 15 15 15 16 16 16 16 15 16 16 16
15 15 15 15</CrACCoeff63>
        </VisualDescriptor>
        <VisualDescriptor xsi:type="ScalableColorType" numOfBitplanesDiscarded="0"
numOfCoeff="256">
```

```
          <Coeff>-206 65 28 50 6 13 20 22 8 10 8 20 7 13 19 16 -5 0 0 3 -1 7 0 0 -6 -1 -5 0 -9 2 2
 -4 -1 -3 0 3 -2 3 4 5 -3 -2 -5 1 -3 0 4 3 1 3 1 7 3 6 2 -1 0 -1 -2 -2 -6 0 -8 -9 0 1 0 0 1 1 2 2 1
 2 2 1 2 2 3 3 -1 -1 0 1 -1 1 1 1 13 2 11 1 5 2 3 -1 0 0 0 1 -1 1 1 2 2 7 2 4 3 5 -1 0 -1 0 -1 1 2 0
 1 0 1 1 1 1 -2 2 -4 -4 0 -1 0 -2 0 -1 1 3 0 -1 0 -3 0 -1 1 2 0 -1 0 -2 -1 -2 2 0 0 0 0 -1 -1 -1 0 2
 0 0 0 0 -1 -1 0 2 -1 -1 0 0 0 0 0 0 0 1 0 2 1 1 1 -1 -1 1 3 1 1 0 1 0 1 -1 0 -3 1 -3 -1 2 0 -1 0 -7
 -1 -2 -1 2 0 -1 1 -7 -1 5 -1 3 0 -3 0 -4 -1 2 -4 6 -1 -1 1 -2 -2 -2 -2 3 0 -1 0 -2 1 -3 1 2 0 -1 -2
 -3 -1 3 0 4 0 -3 -2 -3 -1 3 2 2</Coeff>
        </VisualDescriptor>
        <VisualDescriptor xsi:type="EdgeHistogramType">
          <BinCounts>1 7 6 3 4 3 6 6 5 3 3 5 5 6 5 1 4 5 4 7 2 4 5 6 6 2 5 5 6 5 3 3 5 6 6 4 3 5 3
 7 4 4 5 4 6 2 4 5 6 6 4 3 5 5 6 3 5 5 4 6 2 7 3 3 4 6 6 5 2 2 6 5 4 5 1 7 5 2 1 2</BinCounts>
        </VisualDescriptor>
        <VisualDescriptor xsi:type="DominantColorType">
          <SpatialCoherency>0</SpatialCoherency>
          <Value>
            <Percentage>8</Percentage>
            <Index>8 18 27</Index>
          </Value>
          <Value>
            <Percentage>7</Percentage>
            <Index>0 2 1</Index>
          </Value>
          <Value>
            <Percentage>2</Percentage>
            <Index>27 25 25</Index>
          </Value>
          <Value>
            <Percentage>5</Percentage>
            <Index>7 7 6</Index>
          </Value>
          <Value>
            <Percentage>2</Percentage>
            <Index>6 13 19</Index>
          </Value>
          <Value>
            <Percentage>3</Percentage>
            <Index>10 12 10</Index>
          </Value>
          <Value>
            <Percentage>2</Percentage>
            <Index>17 14 14</Index>
          </Value>
        </VisualDescriptor>
      </Image>
    </MultimediaContent>
  </Description>
</Mpeg7>
```

L. Interim Report

SCALABLE VISUAL DESCRIPTORS
FOR CONTENT BASED IMAGE RETRIEVAL

BEng Project - Week 12 report

James Singleton

ABSTRACT:
This intermediate report details the progress made on the project "*Scalable Visual Descriptors for Content Based Image Retrieval*". The purpose of which is to investigate the MPEG-7 standard from the Motion Picture Experts Group (MPEG). MPEG-7 is not an encoding standard like the previous ones from MPEG (MPEG-1, MPEG-2 and MPEG-4). It defines how to describe multimedia with metadata i.e. data about data. This project aims to examine the MPEG-7 visual colour descriptors and see if they can be extended and used on scalable images.

1. INTRODUCTION:
MPEG-7 defines a set of low-level visual descriptors for describing digital images [1] and other media. They are widely used in Content Based Image Retrieval (CBIR) systems. The objective of this project is to examine MPEG-7 colour descriptors, incorporate scalability into them and develop a system for retrieving related images from a database. The system will include displaying of the retrieved images and automatic computation of the system performance evaluation metrics. The project will be based on the MPEG-7 eXperimental Model (XM) [1,2].

1.1 SPECIFICATION:
The aim of the project is to design software that can query an image database and retrieve images that match certain colour descriptors. In other words, it will find images that look similar to the query image in the colour domain. The software should be able to rank the images that have been matched in order of relevance. The software should calculate the performance of the results returned in terms of precision and recall. Which can be done using the ground truth of the database. This was input by a person and so will need to be verified manually before hand. Use will be made of the MPEG-7 standard and the existing XM software made for it. The software tools should work on any type of image. The image codec could be scaleable or non-scaleable, lossy or lossless.
A more detailed specification was presented in [3].

2. DETAILS OF THE PROJECT TO DATE:
The MPEG-7 standard, and more specifically the colour descriptors, have been investigated. This is the subject of chapter 13 in [1]. There are 7 colour descriptors defined in the standard. These are:

- The colour space descriptor
- The Dominant Colour Descriptor (DCD)
- The Scalable Colour Descriptor (SCD)
- The Group of Frames or Group of Pictures Descriptor (GoF / GoP)
- The Colour Structure Descriptor (CSD)
- The Colour Layout Descriptor (CLD)

This project only deals with the DCD, SCD, CSD and CLD. GoF is designed for video and the colour space descriptor just defines what format the image is in. E.g. whether it is encoded in Red Green Blue (RGB), luminance and chrominance (YUV or YCbCr), Hue Saturation Value (HSV) or any other colour space. Each descriptor expresses a different aspect of the colour in an image.

There are of course other visual descriptors in the MPEG-7 standard, such as the edge histogram descriptor, but these are not being looked at in this project.

2.1 MPEG-7 EXPERIMENTAL MODEL:

The XM software [1,2] was examined. This is the reference software that was developed as a prototype implementation along with the standard. It comes in source code form only and is written in the C++ programming language. It was discovered that the latest code from the Concurrent Versions System (CVS) repository had changed very little from the last snapshot [2] (version 6.1 – 01/01/05), which is also on the DVD that comes with [1]. With regard to the files that this project uses, it was identical. The XM was compiled with the required external dependences and then tested. This required compiling the libraries that the XM software uses to import different image types and format the output files. The resulting Dynamic Link Library (DLL) files had to be put in directories with specified names and included in the project resources. The XM software was compiled and linked to create an executable binary file.

The XM is executed on the command line like so:
XMMain –p ParFile.par *(other optional parameters)*
ParFile.par is a plain text file containing the configuration data. The instructions in the file can be overridden with other command line parameters.

The application that the XM is to use must be specified in the *.par file or on the command line. This defines which descriptor it will use and whether it performs an extraction (server) or search (client). E.g. inserting "*Application ColorLayoutServer*" into the par file or using the "*–a ColorLayoutServer*" flag when launching the XM will extract the colour layout descriptor. A file with a list of images to be processed and the desired output file must also be specified in a similar manner.

The only documentation found about how to configure the XM consisted of brief comments in the parameter files [1,2]. There was some mention in [4], which this project builds upon, but it was vague. Attempts were made to contact the author of [4] but no response has yet been received.

4 colour descriptors (scalable colour, dominant colour, colour layout and colour structure) were extracted for an image into eXtensible Markup Language (XML) [5] files. XML was chosen, as the files are easy to read by hand and simple to process with software. Putting "*CodingMode 1*" into the par file instructs the output to be formatted as XML. This was not very well documented but was discovered through trial and error. There are 2 other output formats supported including a compact binary representation.

The results of the extraction did not seem convincing. Various parameters were tried in the .par files to correct this, but the output file always contained the same values. The XM could not be configured to give correct results, or for some descriptors, even plausible numbers. One of the main reasons for this is the almost complete lack of documentation and that the XM code is very complex. It was written by multiple authors and then stitched together. This makes it difficult to understand, let alone modify. The XM was deemed unsuitable for the intended purpose and an alternative was sort.

2.2 IMAGE DATABASE:

An image database [6] that contained existing ground truth data was obtained. This is the same database as used in [4]. A sample of the data was checked to ascertain its validity. The ground truth was deemed to be correct but also limited. The limitations appear to be due to the subjective nature of the human analysis. All the elements in a picture were not always listed. Most of the major parts were described, e.g. monkey, tree and sky. Often in a complex picture minor elements were omitted e.g. puddle. Two different pictures of trees may be quite

different in colour. It is hard to tell a computer what a tree looks like, as they do not "think" in the same way as people. This difference between human and machine perception is known as the semantic gap. It should still be possible to use this information to gauge the performance of the matching component automatically.

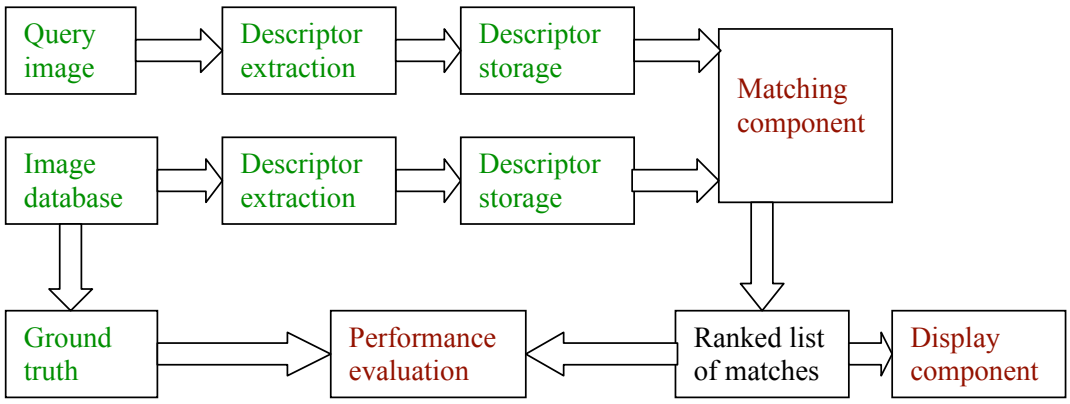
2.3 ALTERNATIVE SOFTWARE:

A suitable alternative piece of software [7] was found. It is a prototype rewrite of some of the XM features. It is open source and licensed under the GNU General Public License (GPL). Caliph [7] is designed for photo annotation and can be used to extract 3 of the colour descriptors to an XML file. Unfortunately it does not support the colour structure descriptor. It has a Graphical User Interface (GUI) and proved to be very easy to use. It is written in the Java programming language. An advantage of using Java is the cross platform portability. It will run on any platform that has a Java Virtual Machine (JVM). In other words, it should work on Microsoft Windows, Linux, Sun Solaris and Apple Mac OS X without modification. Although it runs slower than native code this should not cause problems. The University super computer (iceberg) could be used if required. Included in the Caliph binary is another program called Emir. This allows searching of an image database. Moving to the Caliph and Emir (C&E) project required learning the Java programming language. Java syntax is similar to that of C and it also includes object-oriented design like C++.

An extension of the C&E project [7] is an application library [8]. This takes the core functionality of C&E and allows others to incorporate it into their software. Lire [8] only supports the colour layout and scalable colour descriptors. The dominant colour descriptor is not fully implemented. It uses a database [9] for the descriptor storage, not XML [5]. Another option is to use the XML files generated by [7]. The author of the software was contacted and an encouraging response was received. It will still be possible to investigate the scalability of some of the descriptors. The implementation of the matching component will become more complex. Less code can be reused from [7] and [8] than if the XM had functioned correctly and been used as the code base.

2.4 SYSTEM OVERVIEW:

Fig. 1 illustrates the proposed form the complete system will take. The extraction of the descriptors does not have to take place at the time of query. This indexing can be done prior to the query and stored in XML files or other database.



[FIG. 1 SYSTEM LAYOUT FLOWCHART]

3. REVISION OF THE INITIAL PLAN GANTT CHART:

As of the time of writing (week 11) the time plan has been adhered to.
The matching algorithm is a work in progress as some difficulties have been encountered. The image display and performance evaluation components will have to be pushed back until after the examination period (week 13). The investigation into scalability has yet to start. This is scheduled for after the winter break as well.

Week #s =>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1																		
2																		
3																		
4																		
5																		
6																		
7																		
8																		
9																		
10																		

^ **Component Headings:** (Rows)

1. Research MPEG-7 colour descriptors and XM software.
2. Initial planning.
3. Write and test the descriptor extractor component.
4. Write and test the descriptor storage component.
5. Write and test the matching algorithm component.
6. Run queries with the finished algorithm.
7. Write and test the image display performance metrics evaluation component.
8. Extend the descriptor extractor component to handle scalable images.
9. Test the program retrieval performance.
10. Write week 12 report.

4. CONCLUSION:

A good understanding of the MPEG-7 standard and related technology has been gained. A component for extracting visual descriptors from images and storing them has been made. The matching component is being worked on.

5. REFERENCES:

[1] B.S. Manjunath, P. Salembier and T. Sikora,
"*Introduction to MPEG-7: Multimedia Content Description Interface"*. (Wiley, 2002).
[2] Institute For Integrated Systems, Munich University of Technology,
"Sources for MPEG-7 XM Software",
www.lis.e-technik.tu-muenchen.de/research/bv/topics/mmdb/e_mpeg7.html
[3] J. Singleton, "BEng Project Initial Plan", (University of Sheffield, October 2006).
[4] H. Ding, "Compressed Domain Analysis of Scalable Coded Video Content", M.S. thesis,
(University of Sheffield, 2006).
[5] The World Wide Web Consortium, "Extensible Markup Language", www.w3.org/XML
[6] Department of Computer Science and Engineering, University of Washington,
"Object and Concept Recognition for Content-Based Image Retrieval", "Ground Truth Database",
www.cs.washington.edu/research/imagedatabase
[7] Mathias Lux, Institute for Knowledge Management, Graz University of Technology,
"Caliph & Emir," www.semanticmetadata.net/features/.
[8] Mathias Lux, Institute for Knowledge Management, Graz University of Technology,
"Lire," www.semanticmetadata.net/lire/.
[9] The Apache Lucene project, The Apache Software Foundation, lucene.apache.org