Query by Fax for Content-Based Image Retrieval

Mohammad F. A. Fauzi and Paul H. Lewis

Intelligence, Agents and Multimedia Group,
Department of Electronics and Computer Science,
University of Southampton, UK.
{mfaf00r,phl}@ecs.soton.ac.uk

Abstract. The problem of *query by fax* (QBF) and by other low quality images for art image retrieval is introduced. A reliable solution to the problem, the slow QBF algorithm, is presented but it is too slow for fast interactive retrieval situations. It is used as a yardstick for assessing the quality of a much faster technique, the fast QBF algorithm, based on the pyramid-structured wavelet transform (PWT). Both algorithms use the fact that the fax images are almost binary in nature. In the fast QBF method, the query (fax) image is used to identify what would be an appropriate threshold for the database image. This threshold value is then used to select an appropriate set of precomputed wavelet signatures to be used for matching the query and database image. Results of the algorithms are compared. The speed and accuracy of the new technique make it a valuable tool for the *query by fax* application.

1 Introduction

The motivation for this research comes from a requirement by some museums to respond to queries for pictorial information which are submitted to them in the form of fax images or other low quality monochrome images of works of art. The museums have databases of high resolution images of their artefact collections and the person submitting the query is asking typically whether the museum holds the art work shown or perhaps some similar work.

Typically the query image will have no associated metadata and was produced from a low resolution picture of the original art work. The resulting poor quality image, received by the museum, leads to very poor retrieval accuracy when the fax is used in standard *query by example* searches using, for example, colour or texture matching algorithms. Some examples of genuine fax images received by the museum together with the corresponding high resolution images they represent are shown in Fig. 3. Both good and poor quality faxes are shown for illustration, and will be used in our experiments. In this paper, we make the assumption that query images represent the full original and are not partial image queries.

To increase the speed of retrieval, most content-based retrieval systems use feature vectors as a medium for the similarity match. The feature vectors of all images in the database are computed and stored in advance in a unique feature vector database. During the retrieval process, the feature vector of the query image is computed and is compared to all the feature vectors in the database. Images are retrieved based on the distance measure between query and database image feature vectors.

Recently, wavelets have been used in many areas in image analysis. They deliver a precise and unifying framework for the analysis and characterization of a signal at different scales, hence they provide an excellent tool for multiresolution analysis. In [1], Mallat proposed the pyramid-structured wavelet transform (PWT), a computationally very fast method used widely in image analysis and matching.

In this paper we show that poor retrieval performance in *query by fax* can be overcome by using a modified approach to wavelet-based retrieval. Since the quality of the fax image is so low that it differs substantially from its original, applying the PWT to the original image will not produce a feature vector close enough to the feature vector from the query. The fax image is first converted to a binary image before the PWT is applied. A similar conversion to binary is applied to each of the database images, choosing a threshold which makes them as close as possible to the binary fax image, before the PWT is applied.

At first, this method seems to be unsuitable for use in the feature vector database approach mentioned above, since it requires the feature vectors of the images in the database to be computed in advance. However we explain in section 3 how we can still use the same concept for this particular method, due to the compact nature of the wavelet signatures. Once we have created the binary images, the feature vectors are computed using the PWT, and the ones nearest to the query are selected.

The algorithm, which we call the fast QBF method, has the advantage of high retrieval accuracy while maintaining a low computational load. In section 4 we compare the performance of the algorithm with a slow QBF method that gives very high retrieval accuracy but which is computationally very intensive. The fast QBF algorithm almost matches the high accuracy of the slow QBF method but with a much lower computational load.

This paper is organized as follows. The next section will briefly review the wavelet transform algorithms to be used as part of our approach. Section 3 describes the stages of the fast QBF method and in section 4 the method is evaluated and compared with both a basic PWT algorithm and the slow but reliable QBF method. Finally, the conclusions are presented in section 5.

2 Review of wavelet transform

Wavelets were invented to overcome the shortcomings of the Fourier Transform. The fundamental idea is not only to analyse according to frequency but also according to scale, which is not covered by the Fourier transform. Most standard wavelets are based on the dilations and translations of one mother wavelet, ψ, which is a function with some special properties.

In image processing terms, the dilations and translations of the mother functions ψ are given by the wavelet basis function:

$$\psi_{(s,l)}(t) = 2^{-s/2}\psi(2^{-s}x - l).$$

The variables s and l are integers that dilate and orientate the mother function ψ to generate a family of wavelets, such as the Daubechies wavelet family [3, 7]. The scale index s indicates the wavelet's width, and the location index l gives its position in 2-D.

Wavelets can be divided into orthogonal and non-orthogonal. They are orthogonal if their basis functions are mutually orthogonal. Because of the orthogonality, there exist a series of coefficients that can represent the wavelet decomposition for the whole family of a particular wavelet. The series of coefficients is named the Quadrature Mirror Filter (QMF). This concept, along with the theory of filter banks, is the basis for producing the famous fast algorithm PWT decomposition by Mallat [1].

Quadrature mirror filter is a filter that can be either low-pass or high-pass just by changing sign and rearranging its coefficient. In 2-D wavelet decomposition, the QMFs are used as both highpass and lowpass in both horizontal and vertical directions, followed by a 2 to 1 subsampling of each output image. This will generate four wavelet coefficient images, i.e. the *low-low*, *low-high*, *high-low* and *high-high* (*LL, LH, HL, HH* respectively) subbands. The process is repeated on the *LL* channel, and the 2-D PWT decomposition is constructed. Fig. 1(a) illustrates the PWT decomposition up to 3 levels, while Fig. 1(c) shows the result of PWT decomposition (also referred to as the wavelet coefficients) on the image in Fig. 1(b). For detailed information on wavelet theory, please refer to [2, 4, 5].

In this paper, the PWT wavelet coefficients are used to compute the feature vector of an image. The information obtained from this is of great importance as it shows the dominant frequency channels of a particular image and is particularly useful in similarity matching.

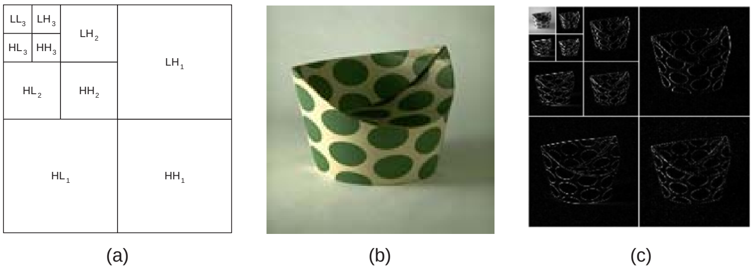

Fig. 1. (a)Wavelet decomposition using PWT. (b)An image. (c) Wavelet decomposition of image in (b)

3 Similarity retrieval using fax images

In this section we describe the various stages of the fast QBF algorithm. The binary image thresholding technique and the PWT feature vector computation is discussed, along with all the necessary requirements for performing an image retrieval task.

3.1 Binary image threshoding

As stated earlier, since the query images are almost binary, it is better to compute feature vectors in the binary domain. The query image can be converted to binary by thresholding in the middle of the grey scale range covered by the image. In order for the feature vector of a database image to be compared fairly with the feature vectors from the query, the database image must also be converted to binary. But the choice of threshold is not immediately obvious. For the original database image corresponding to the query, an appropriate threshold is probably one that produces a binary image with the same percentage of black (or white) pixels as the binary form of the query image. We could use this percentage for all the database images but it varies from query to query.

How do we match the percentage if we are to precompute the feature vectors from the binary versions of all the database images? Note that since the query image is the target and already effectively binary, it is the original image that must be made as close as possible to the binary query and not vice versa. One way to solve this problem is to convert each database image into a set of different binary images corresponding to different percentages of black pixels from 0 to 100%. If sufficient binaries are created, the binary query image will then be very similar to one of these binaries for the original image. We chose to create 99 binaries for each database image corresponding to percentages of black pixels from 1 to 99 in steps of 1%.

However, we do not need to store the binaries. Calculating the feature vectors for the database involves calculating the PWT for each of the binary images for each image in the database. This is implementable since the PWT is a fast algorithm and, more importantly, the feature vectors for each binary image have only a relatively small number of coefficients. During matching, only one of the sets of wavelet coefficients will be used for each database image, namely the set associated with the same black pixel percentage as the binary query image.

3.2 Feature vector computation and comparison

As described earlier, the PWT is used as a tool for feature vector extraction. Since the PWT can be applied only on dyadic square image, the binary images are all resized to 256×256. The resizing can also be done before the binary conversion. The PWT algorithm is applied and the image is decomposed into 4 subimages or subbands (LL,LH,HL and HH). The LL band is decomposed further until the smallest subimages are of size 4×4, i.e. 6 levels of decomposition. This results in 19 different subimages or subbands.

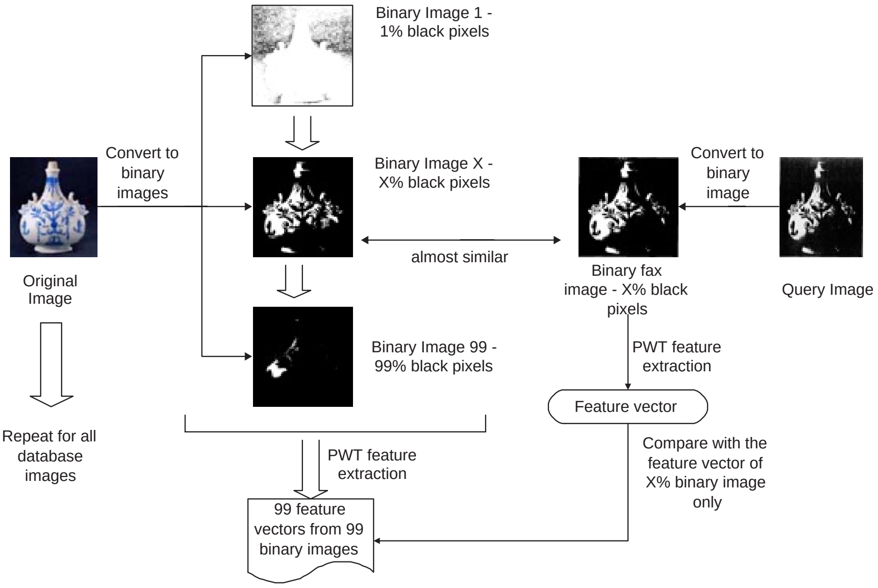

Fig. 2. Binary image matching between fax image and its original

Once the wavelet coefficients of a binary image are available, features are computed from each subband, resulting in 19 features for each binary image. The mean, μ is the energy measure used to compute the features. Let the image subband be $W_{mn}(x,y)$ and mn denotes the specific subband, m is the decomposition level and *n=1,2,3,4* indicates the *LL, LH, HL, HH* bands respectively, then the μ_{mn} is calculated by:

$$\mu_{mn} = \frac{1}{N_{mn}^2} \int\int |W_{mn}(x,y)| \, dx \, dy.$$

where N is the length of a particular subband mn. The feature vector f for a particular binary image is therefore:

$$f = [\mu_{mn}]\,,\ n \neq \mathit{1}\ \ \textit{except for the coarsest level, i.e. m=6,}$$

$$= [\mu_{12}, \mu_{13}, \mu_{14}, \mu_{22}, ...\mu_{61}, \mu_{62}, \mu_{63}, \mu_{64}]$$

The feature vectors for the database images will have *99x19=1881* coefficients, although only 19 will be used for comparison in each retrieval task. The distance classifier used is the Euclidean minimum distance. The distance between 2 features i and j is given by:

$$d(i,j) = \sqrt{\sum_m \sum_n \left[\mu_{mn}^{(i)} - \mu_{mn}^{(j)}\right]^2}$$

Once the distances are computed, the images will be retrieved in order of increasing distance from the query image.

4 Performance evaluation

Experiments have been conducted using each of 20 genuine fax images as the query, and a database consisting of 1058 images of various types and sizes, including the original images of the 20 fax images. The fax images and their originals are shown in Fig. 3. The evaluation is based on the ability of the algorithm to retrieve the original image when the fax version of the original is used as the query. The results for the fast QBF algorithm in table 1 show the retrieval position of the original image among the 1058 database images, using *Daubechies* family wavelets with 8 vanishing moments as the wavelet bases.

The table also shows the results obtained by using a basic *query by example* retrieval with the same PWT features but calculated from the raw query and database images without the initial binarisation stage. It can be seen that the basic *query by example* algorithm is particularly poor for fax queries, but the retrieval results obtained using the fast QBF algorithm are very encouraging. All the original images are retrieved within the top 5. This is a good result considering the poor quality of fax images and the reasonably large image database used. The results suggest that the distance between the fax images and their originals are very close and should still produce good results for a larger image database. Different wavelet bases were also tested in this experiment, and it was found that the choice of wavelet base (*Haar* [7] and *Battle-Lemarie* [7] family) has little effect on retrieval result. However the *Daubechies* wavelet gives a slightly better result, probably because it is compactly supported in both the time and frequency domain [6].

To further compare the performance of our fast QBF algorithm, we implemented an alternative slow QBF method for *query by fax*. However, this algorithm is not appropriate for interactive retrievals since it does not use feature vectors as a medium for comparing images but uses pixel by pixel comparison resulting in a very high computational load. It is not possible to do any precomputing without using large amounts of storage for all database images so all calculations are performed during the retrieval process. Hence it is a very slow method. The method is used as a yardstick for comparison since, although computationally intensive, it achieves very high accuracy retrieval.

The slow QBF method is similar to the fast QBF algorithm except that, instead of using the PWT on the binary image, a pixel by pixel comparison is made between the binary fax and the binary of the database image with the appropriate percentage of black pixels. Both binaries are resampled to the same size (*64x64* pixels). The number of matching pixels, as a percentage of the total number of pixels, is used as the similarity measure and the database images are then retrieved in decreasing order of similarity.

Table 1 also shows the results for this slow QBF method. As expected, the slow QBF algorithm gives very good retrieval results. Almost all the originals

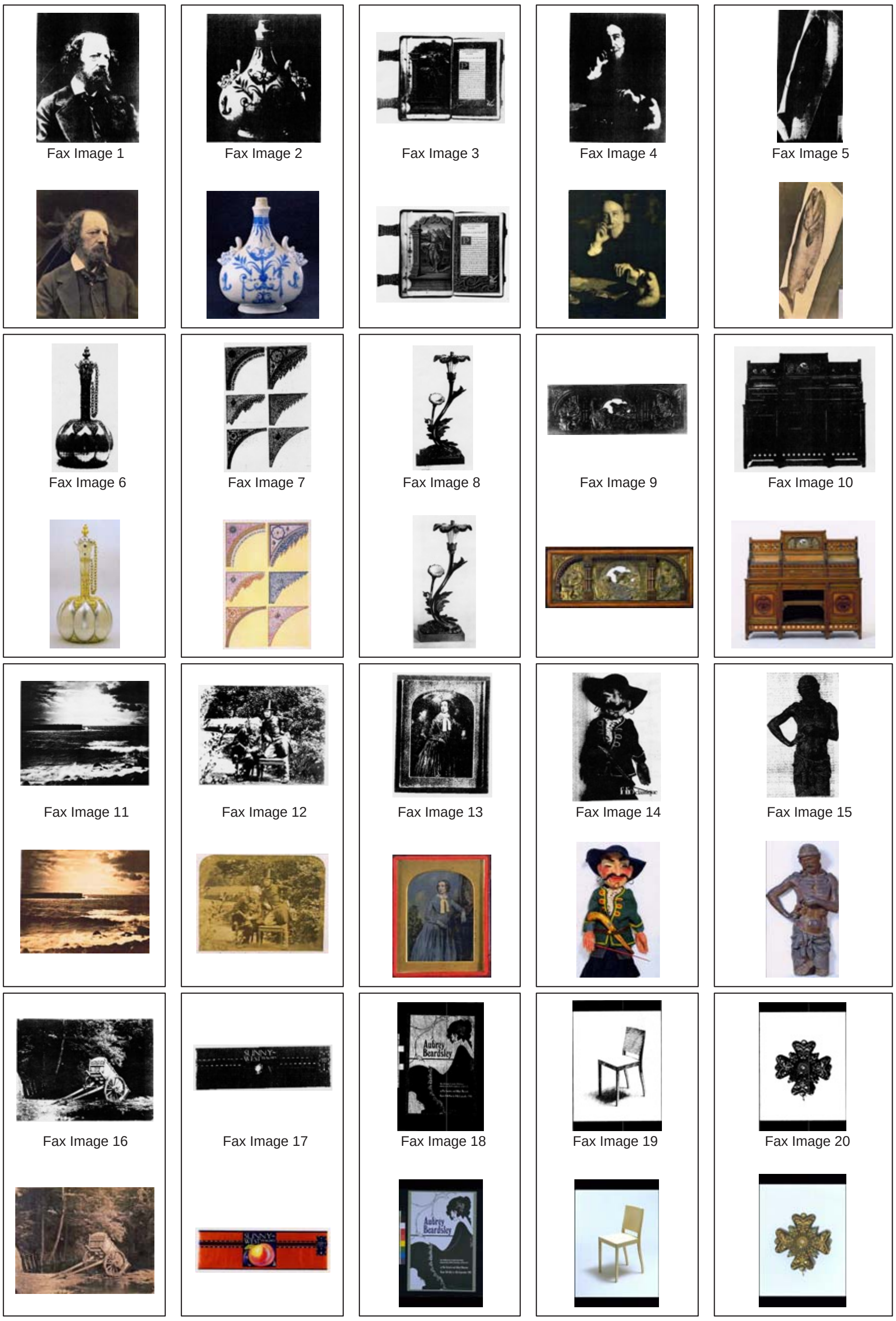

Fig. 3. The fax images used in this experiment and their originals

Table 1. Retrieval results using 20 fax images on database of 1058 images

Query Image No.	Rank of Original		
	Basic PWT Technique	Slow QBF Technique	Fast QBF Technique
1	104	1	1
2	369	1	1
3	15	1	1
4	21	1	3
5	272	1	1
6	130	1	1
7	258	1	1
8	2	1	3
9	502	1	1
10	302	20	2
11	603	1	1
12	299	1	1
13	60	1	1
14	495	1	4
15	500	1	2
16	339	1	1
17	15	1	2
18	264	1	4
19	1	1	1
20	1	1	1

are returned as the first match. Table 2 compares the average time taken for retrieving images from the database of 1058 images with the basic PWT algorithm and the slow and fast QBF algorithms. The times are for a 700 MHz Xeon processor. From table 1 and table 2 it can be seen that the fast QBF algorithm almost equals the slow QBF method in terms of retrieval performance, but involves a much smaller computational load. In other words, we can say that the fast QBF method integrates the high accuracy of the slow QBF method with the low computational load of the basic PWT method.

5 Conclusion

The main aim of this paper was to investigate the possibility of retrieving similar images from a large database using low quality monochrome images such as fax images as the query, with speed and high accuracy as main objectives. The wavelet transform, combined with the multiple binary thresholding and matching method provides an appropriate platform to tackle this problem, and the objectives have been achieved successfully. Also, the main purpose of the algorithm is to find either the exact parent image or similar images in the database. So in our application, the confusion level in the database is not a crucial factor.

Table 2. Comparison of speed between the three algorithms

Time taken to retrieve images (in seconds)	Basic PWT Technique	Slow QBF Technique	Fast QBF Technique
	1	130	1

None of the algorithms presented attempt to address partial (sub-image) queries and the issue of rotation and translation invariance. We are currently working on the possibility of partial query handling, as well as rotation and translation invariant versions to improve the system, both for *query by fax* and for general retrieval applications.

6 Acknowledgments

The first author is grateful to the Department of Electronics and Computer Science for financial support. We are also grateful to the EU for support under grant number IST_1999_11978 (The Artiste Project), and to one of our collaborators, the Victoria and Albert Museum, for use of their images.

References

[1] Mallat, S.: A theory for multiresolution signal decomposition: The wavelet representation. In IEEE Transactions on Pattern Analysis and Machine Intelligence. **11(7)** (1989) 674-693

[2] Daubechies, I.: The wavelet transform, time-frequency localization and signal analysis. In IEEE Transactions on Information Theory **36(5)** (1990) 961-1005

[3] Graps, A.: An introduction to wavelets. In IEEE Computational Science and Engineering **2(2)** (1995) 50-61

[4] Ngunyen, T., Gunawan, D.: Wavelet and wavelet-design issues. In Proceedings of ICCS **1** (1994) 188-194

[5] Mallat, S.: Wavelets for a vision. In Proceedings of IEEE **84(4)** (1996) 604-614

[6] Nason, G.: A little introduction to wavelets. In IEE Colloquium on Applied Statistical Pattern Recognition (1999) 1/1-1/6

[7] Wavelab Toolbox ver. 802 for Matlab. At http://www-stat.stanford.edu/ wavelab/