An Integrated Content and Metadata based Retrieval System for Art

P.H.Lewis, K.Martinez, F.S.Abas, M.F.A.Fauzi, S.C.Y.Chan, M.Addis, C.Lahanier, J.Stevenson

Abstract— **In this paper we describe aspects of the Artiste project to develop a distributed content and metadata based analysis, retrieval and navigation system for a number of major European Museums. In particular, after a brief overview of the complete system, we describe the design and evaluation of some of the image analysis algorithms developed to meet some specific requirements of the users from the museums. These include a method for retrievals based on sub images, retrievals based on very low quality images and retrieval using craquelure type.**

***Index Terms*— Content based image retrieval, art images, wavelets, crack analysis**

I INTRODUCTION

The aim of the ARTISTE project was to develop an Integrated Art Analysis and Navigation Environment hosted on a distributed database and accessed via the World Wide Web. The environment includes a range of facilities for searching and analysing digital art images and includes tools not only for retrieval using metadata but also for a wide range of content based retrieval activities. Partners in the collaboration included NCR Systems Engineering Copenhagen, Giunti Interactive Labs, Centre of Research and Restoration of the Museums de France at the Louvre, the National Gallery London, the Uffizi Gallery in

Manuscript received January 22, 2003.
This research was partly financially supported by the European Commission's Fifth Framework programme, project Artiste.: IST-1999-11978. Server equipment was kindly provided by Hewlett Packard's Art & Science programme.
P. H. Lewis, K. Martinez, F. S .Abas, and M .F. A. Fauzi are with the Electronics and Computer Science Department of the University of Southampton, UK. phone: +44 (0)23 80594491; fax +44 (0)23 80592865;
e-mail: [phl,km,fsa00r,mfaf00r]@ecs.soton.ac.uk.
M. Addis is with the IT Innovation Centre, Southampton, UK.
C. Lahanier is with the Centre de Recherche et de Restauration des Musées de France, Paris, France.
J. Stevenson is with the Victoria and Albert Museum, London, UK
S.C.Y.Chan is with Panasonic, Basingstoke, UK.

Florence, the University of Southampton and the Victoria and Albert Museum London.

Content-based retrieval (CBR) of art images has been a continuous challenge [1]–[3]. In this research area there are often weakly defined requirements from real users and common image processing algorithms are often applied and their success measured. In Artiste we attempted to target individual retrieval tasks which were provided by our museum partners in order to obtain more satisfactory results. Some tasks required new algorithms while others could start from fairly conventional and well understood algorithms. They also fell into two types: providing a list of images ranked by score or producing new metadata which would be stored once only. A list of problem areas was initially established through discussions with the museum partners. Particularly useful problems were identified by direct interviews with museum staff that spent most of their time searching for images in their picture library. More complex tasks were from conservation researchers that needed tools to help them explore their data in a way that would normally take too long (such as a statistical analysis of cracks across their collection). The resultant list of image-related user requirements included:

- Finding similar images – especially “do you have this image?”
- Searching using synonyms and multilingual vocabulary
- Searches based on the restoration framework (eg cracks, wooden structures)
- Colour-based searches
- Query by sketch
- Find the parent image of a detail
- Query by Faxed images
- Query by Border shape

After a brief overview of the architecture and the general facilities provided by the system, this paper will concentrate on some specific problems for art image retrieval and analysis which were identified by the galleries and addressed in the project. These include retrieval using sub-image queries, retrieval when only a low quality example query image is available and identification and classification of craquelure.

II System Architecture

ARTISTE allows combined metadata and image content-based search and retrieval across multiple, distributed image collections. RDF [4] is used to map gallery-specific textual metadata schema to common standards such as Dublin Core [5].

RDF schema is also used to describe image content as a metadata item. The operators (implemented using image processing algorithms.) relating to image content are explicitly defined along with the constraints under which these operators can be applied. For example, the MCCV algorithm implements a 'part of' operator which can be applied to an image to allow a sub-image to be found within it. The textual metadata equivalent would be to search within a string to find a key word. In this way it can be seen that a uniform approach is taken to the semantics of searching textual content and image content.

Furthermore, the use of RDF provides a flexible solution to cross-collection searching. Mapping to common semantics requires no changes to local metadata and schemas. Multilingual translation of metadata attribute names allows the user to use their native language when specifying which attributes to search over for multiple collections. Finally, by publication of the RDF, a gallery can provide a machine readable description of any search and retrieval services it chooses to expose. This allows software clients to dynamically constrain the search facilities offered to users so that only the available query functionality is used.

With over 2 Tbytes of images available within the consortium, scalability was a prime concern. Images accepted into the systems are *analysed* by algorithms to produce *feature vectors* which are stored as binary large objects (BLOBS) within the database. During a query, these feature vectors are compared by fast operators which return scores. Although indexing could have produced considerable speed increases in some cases, this has not yet been implemented as effort has been concentrated on algorithm development. All the image processing code was written in C/C++ using the VIPS/ip library developed in past projects [6]. The NCR TOR database provided automatic symmetric parallelism to further speed-up queries [7] although this has since been replaced with mySQL [8]. An administration interface also allowed new algorithms to be added to the system.

III Sub-image Matching

Many content-based image retrieval systems work with whole images and address questions of the form “find me images similar to this query”. General techniques based on such features as colour distribution, texture, outline shape and spatial colour distribution have been popular in the research literature and in content based retrieval systems. Several extensive general reviews of content-based retrieval (CBR) techniques have appeared in recent years [9]. CBR has been applied to art images since the 1980s with the Morelli project [1] and IBM's QBIC for example has been applied to such images in collaboration with UC-Davis [2] Previous approaches usually applied a generic CBR system to a group of Art images to see how useful it would be. In Artiste, by contrast, our aim has been to solve specific CBR problems by developing algorithms tailored to these tasks, rather than producing a generic solution for all queries. The use of the colour histogram [10] for comparing images has been popular, primarily because it is easy to compute, is fairly insensitive to small variations between images and is relatively efficient in terms of computation speed and storage requirements. It has the disadvantage that it does not capture any information about the distribution of the colours spatially within the image and various techniques to capture such information have been proposed including the colour coherence vector approach [11] and the use of colour pairs [12].

Many techniques like these only work on the complete image. Hence they will not allow the query image to be a sub-image of the image to be retrieved and require similarity in image resolution between query and target. One of our collaborating galleries had a specific requirement to be able to retrieve the full image of a painting, given a query image which is of a fragment of the full painting (i.e. a sub-image) and the query may have been captured under different capture conditions and possibly at a different stage in the restoration process from the parent. In this section we present an approach to sub-image retrieval for museum collections which include large numbers of very high resolution art images [13].

Our approach to the sub-image problem has been to divide each database image into a pyramid of image patches, extracting feature vectors for each patch and matching the query feature vectors with each of the feature vectors from the database images to find the best match. We have used a patch size of 64 by 64 pixels, a compromise reached by experimentation, and not only tile the whole image with such patches but also repeat the process with another set of patches offset horizontally by 32 from the first set, then offset vertically by 32 and finally offset both horizontally and vertically. The resolution of the image is then reduced by a factor of two and the process repeated recursively until the image is approximately covered by a single patch. In general, global techniques such as the colour histogram are not effective for sub-image matching in their basic form and previous workers have also used the hierarchy approach.

In our approach, for matching based on colour, we use the colour coherence vector (CCV) rather than the basic colour histogram used in earlier decompositions, as the representation of the individual image patches as it carries useful local spatial information. The colour coherence vector [11] records the numbers of coherent and incoherent pixels of each colour in the patch, where a coherent pixel is one which belongs to a region of similar coloured pixels whose relative size is above a certain threshold. The colour space is quantised into 64 cells. The CCVs are coded for rapid matching and can be compared at an average rate of 265,000 per second on a 600MHz Pentium III processor.

In figure 1 we show a query image which is a fragment of the Moise présenté à Pharaon by Orsel Victor captured before restoration work at the Louvre gallery. The best match is shown in the same figure and it can be seen that this is the correct parent image but after restoration. (The before restoration image was not in the database). Note that the position of the match is highlighted, a useful facility when dealing with small fragments from ultra high resolution images. This particular parent image is 6328 by 4712 pixels. It should be stressed that the parent image was scanned at 1.47 pixels per mm of the original painting whereas the sub-image query was captured at a quite different resolution, 0.92 pixels per mm. In this test there were over 1000 images in the database varying in size from 440,000 pixels to 30,000,000 pixels and the retrieval process took about 45.8 seconds on a Pentium III 600Mhz PC.

In figure 2 we show a further example of the multi-scale CCV retrieval in action. The query image is a selection from a fabric from the Victoria and Albert collection and the best retrievals show the parent tapestry and others containing a similar motif to the sub-image query pattern. The retrieval process took 13.74 seconds to complete for the same test database.

An Evaluation of the MCCV Algorithm

In order to evaluate the performance of the MCCV algorithm when sub-image quality is less than ideal we use the test database of 1000 images described above. Sets of query sub-images were selected at random from the test database and degraded in a variety of ways and with varying strengths. Eight strength levels were used for each corruption type. The three types of corruption were:

- **Blurring:** which consisted of the application of a 3x3 averaging window, applied once for the first strength of blur, twice for the second and so on.
- **Noise addition:** which consisted of adding Gaussian noise to the sub-images. The strength of the noise was determined by the standard deviation, sigma, of the Gaussian noise distribution. A strength of one corresponds to a sigma of one, a strength of two corresponds to a sigma of two and so on.
- **Resizing:** which consisted of reducing the size of query sub-images by a specified percentage determined by the strength. The strength of one corresponds to 0% reduction in image size, the strength of two corresponds to 10% reduction, the strength of three for a 30% reduction and so on.

The evaluation procedure was as follows:

Randomly select 100 images, *I*, from the database.

For each image in I randomly select a random sized sub image area and store in K

For all image operations, *O* (Blur, noise, sub-sample, identity)

For each image I_n in *K*

Apply image processing operation *O* to image I_n to produce image *J*.

Create feature vector *F* from image *J*.

Compare *F* with all feature vectors in database.

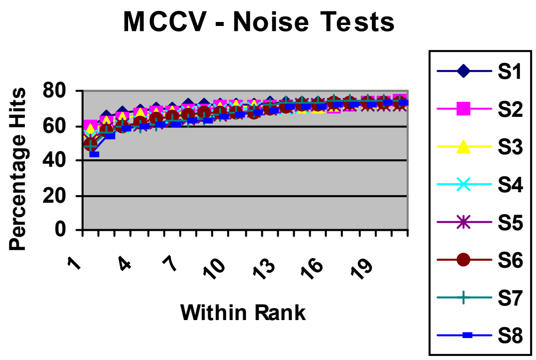
Order similarities and record rank information.

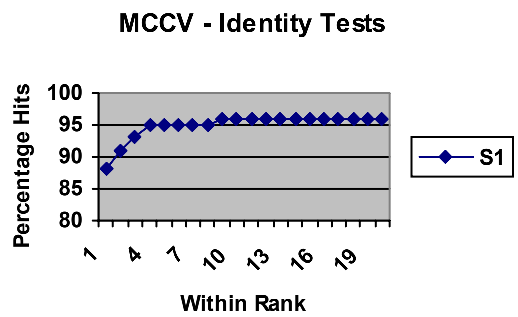
The purpose of these tests is to estimate how well the algorithms cope with different forms of image corruption under different degrees of severity. The tests should indicate how the algorithms might perform in real-world applications.

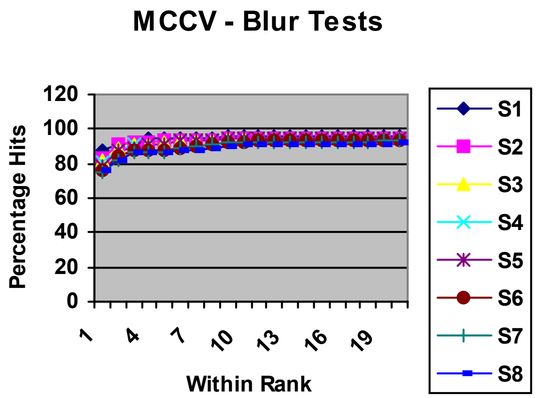
B Results of Evaluation

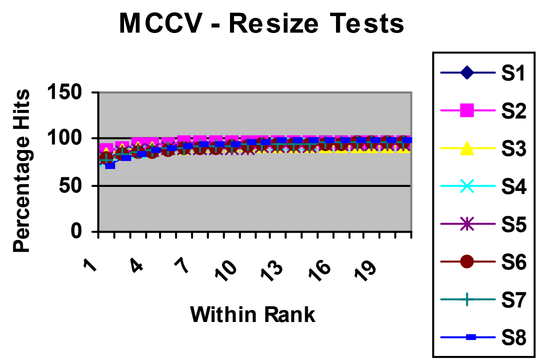
The graphs presented below are cumulative graphs of the retrieval results. They show the percentage of correct matches, across the tests, found within the top N hits where N is given by the within rank value on the x coordinate. The legends in the graphs show the strengths at which the algorithms were applied, where S1 is the weakest through to S8 the strongest.

The first graph below shows the results when no corruption is applied to the sub-images. It can be seen that 88% of the tests gave the correct match as the best match found and 96% of the tests found the correct match in the top ten best matches. The Blur test shows that even with the highest strengths of blur, the correct matches are found above 90% of the time within the top 10 hits. The robustness to noise is less successful and the correct matches are found in the top 10 hits for only about 70% of the tests. Robustness to resizing is reasonable since, although only about 70% of the tests gave the correct match as the best match when applying the strongest resizing, it can be seen that at least 90% of the tests find the best match in the top ten for all strengths of resizing.









C Generic Multi-scale

The multi-scale algorithm has been implemented (in C++) in such a way that any appropriate algorithm applicable to whole images, may be used without further implementation in a multi-scale version. Thus, for example, our texture matching algorithm, based on the wavelet transform, is also used for sub-image matching when the required basis for sub-image matching is texture. The result is a binary array of feature vectors and each pair can be compared using the same matcher as for whole images, thus reducing development effort.

IV Query by Low Quality Image

One of the problems which galleries sometimes encounter is being asked "have you got this particular painting or art work in your collection?" when the image submitted as the basis for the query is a very poor and anonymous representation of the original. The image is typically monochrome, perhaps a photocopy of a book image, a fax of a postcard or some other anonymous low quality representation. The query images in these cases are essentially binary and the problem for *query by example* is that, in order to calculate the similarity to a database image, it is necessary to degrade the database image in a way which depends on the query so that pre-computation of the feature vectors is not straightforward.

Two approaches to the problem were developed. Both involve resizing the images to some standard. The first approach uses a pixel by pixel comparison, resulting in very slow but relatively high accuracy retrieval. This method is referred to as the slow query by low-quality image (QBLI) method. It is used as a yardstick for comparison to a faster, wavelet-based algorithm. We call this wavelet-based method the fast QBLI method. It is a modification of the slow QBLI method and has the advantage of high retrieval accuracy while maintaining a low computational load. Throughout this section genuine fax images, received by galleries, are used as our examples of low-quality query images.

A The Slow QBLI Method

The slow QBLI method is based on the fact that the low quality images are almost binary in nature. The query image is first converted to be totally binary, and the percentage of black (or white) pixels is computed. The image to be compared to the query is then converted to binary by thresholding the luminance in such a way that the percentage of its black (or white) pixels is similar to that in the binary query image. Both binary images are re-sampled to the same size 64 x 64 pixels) and a pixel by pixel comparison is made. 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 presented in decreasing order of similarity. This is illustrated in Fig. 3.

The slow QBLI algorithm is not ideal for interactive retrievals since it involves a high computational load. Using this algorithm, it is not possible to do any pre-computing without using large amounts of storage to hold multiple binary versions of each database image, corresponding to the range of possible thresholding values. For this reason, all calculations are performed during the retrieval process making this a slow but, as will be shown later, a very accurate process for query by low quality images.

B The Fast QBLI Method

For this method the Pyramid Wavelet Transform (PWT) coefficients [14] are used to compute the feature vector of an image. The reason for using the PWT is that, for a non-textured image, the frequency content is concentrated in the low frequency region, thus image decomposition is needed just for the low frequency band. Moreover, the PWT has a compact feature vector and a low computational load, which is essential for the retrieval algorithm.

Since the quality of the query images is so low that they differ substantially from their originals, applying the PWT to the original image will not produce a feature vector close enough to the feature vector from the query. As in the slow QBLI method, the query 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 (in terms of the proportions of black and white pixels) to the binary fax image, before the PWT is applied.

At first, this method may again seem to be unsuitable for interactive retrieval applications since it requires the feature vectors of the images in the database to be computed in advance. However, due to the compact nature of the wavelet signatures, it is possible to implement the algorithm in an effective manner. The algorithm consists of two steps; binary image creation and feature vector computation and comparison.

C Binary Image Creation

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 at the centre 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 again chosen as the one that produces a binary image with the same percentage of black (or white) pixels as the binary form of the query image. This percentage could be used for all the database images, but it varies from query to query.

How can the percentage be matched if the feature vectors from the binary versions of all the database images are to be pre-computed? Note that since the query image is the target and already effectively binary, it is the database images that must be made as close as possible to the binary query and not vice versa. A solution to this problem is to convert each database image into a set of different binary images corresponding to different percentages of black pixels between 0 and 100%. If sufficient binaries are created, the binary query image will then be very similar to one of these binaries for the original image. Ninety nine binaries were created for each database image corresponding to percentages of black pixels from 1 to 99 in steps of 1%.

However, the binaries do not need to be stored. 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, for each database image, only one of the sets of wavelet coefficients will be used in the comparison, namely the set associated with the same black pixel percentage as found in the binary query image.

D Feature Vector Computation and Comparison

Since the PWT can be applied only on dyadic square images, the binary images are all resized to 256 x 256. The resizing can also be done before the binary conversion. The PWT algorithm is applied and the image is decomposed into four sub-images

({LL, LH, HL and HH). The LL band is decomposed further until the smallest sub-images are of size 4 x 4, i.e. six levels of decomposition. This results in 19 different sub-images or sub-bands.

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

$$\mu = \frac{1}{N_{mn}^2} \iint |W_{mn}(x,y)| \, dxdy$$

where N is the length of a particular sub-band mn. The feature vector, f, for a particular binary image is therefore:

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

$$= [\mu_{1,2}, \mu_{1,3}, \mu_{1,4}, \mu_{2,2} ...., \mu_{6,1}, \mu_{6,2}, \mu_{6,3}, \mu_{6,4}]$$

The feature vectors for the database images will have 99 x 19=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:

$$d(i,j) = \sqrt{\sum\sum [\mu_{mn}^{(i)} - \mu_{mn}^{(j)}]^2}$$

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

E Performance Evaluation

Experiments were conducted using as the query, each of the 20 genuine fax images with a range of qualities 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. 4. 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 QBLI algorithm in table 1 show the retrieval position of the original image among the 1058 database images, using Daubechies [15] family wavelets with 8 vanishing moments as the wavelet bases. Fig. 5 shows an example of three retrieval results using the fast QBLI algorithm.

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. These are the sort of results one might expect from a standard CBIR system without special algorithms for special cases. It can be seen that the basic *query by example* algorithm is particularly poor for these low quality queries, but the retrieval results obtained using the fast QBLI algorithm are very encouraging. All the original images are retrieved within the top 5. This is a good result considering the poor quality of some of the fax images. The results suggest that the distances between the fax images and their originals in our feature space 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 and Battle-Lemarie family [15]) 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.

	Rank of Original		
Query Image No.	Basic PWT Technique	Slow QBLI Technique	Fast QBLI 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

Table 1: Retrieval Results using 20 fax images on a database of 1058 images

Table 1 also shows the results for the slow QBLI method. As expected, the slow QBLI algorithm gives very good retrieval results. All the originals were returned as the first match, except for one case only, which is because that particular fax image and its original were of different object-to-background ratio, hence resulting in rather dissimilar images after binarisation. 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 QBLI algorithms. The times are for a 700 MHz Xeon processor. From table 1 and table 2 it can be seen that the fast QBLI algorithm almost equals the slow QBLI method in terms of retrieval performance, but involves a much smaller computational load.

It is also important to note that the PWT algorithm applied on binary images helps to minimise computation time. To sum up, it can be said that the fast QBLI method integrates the high accuracy of the slow QBLI method with the low computational load of the basic PWT method giving an effective approach to query with low quality example.

The two approaches to *query by low-quality image* give very good retrieval accuracy, although the speed of the fast QBLI method makes it much more suitable for interactive use. The fast QBLI method illustrates the importance of the wavelet-based feature extractor in this application, where in PWT, we have a very fast algorithm and compact feature vectors. These two constraints are important in using the fast QBLI method where multiple feature extractions as well as multiple feature vectors are necessary for each database image. Other feature extraction techniques are either slow or have large feature vectors.

Time Taken To Retrieve Images (in seconds)	Basic PWT	Slow QBLI	Fast QBLI
	1	130	1

Table 2: Comparison of computation times between the three algorithms

Another important observation from this part of the project is that although we used fax images in the evaluation, method proposed is an excellent way to search using low quality images in general. This is shown by the fact that the basic PWT technique gives very poor retrieval accuracy when used in this kind of problem and suggests that other standard feature extraction methods will also fail to deliver a good performance if not accompanied by a tailored pre-processing stage.

V Query by crack type

A new algorithm is being developed [16] in order to process two unusual query types: “find images with cracks like this image” and “find images with this class of cracks”. The overall goal was to analyse sub-regions of the images in order to classify the cracks (called craquelure) in the paint layer into one of a few classes [17]. Example classes are shown in Fig. 6.

We used a morphological top-hat operator [18], [19] and adaptive [20] grid-based thresholding to detect the cracks. After thinning, the cracks are represented as chain code [21] and statistics gathered on the crack segments. These include lengths, straight line to actual length ratio (LTRL), directionality and orientation histogram [22]. In order to compare two crack networks from a query image for example a distance measure is used based on the two sets of statistics.

Figure 7 illustrates the result of a fuzzy k-means clustering of two of these features. Cluster FKM2 represents cracks which are very directional and straight for example. By combining cluster scores for all the features a crack type score can be obtained. The issue of scale is complex because an area of cracks can cover very large areas of an image. Our current research involves merging the classified regions so that a simpler result can be provided to the user. Linking the cracks to features on the reverse of the painting, from different images, as well as to the overall colour of an area will be the next step in providing a useful research tool for conservators.

VI Conclusions and Future Work

In this paper several of the special purpose algorithms developed for handling particular aspects of art image retrieval have been presented and evaluated. They form part of the Artiste system developed in a European Collaboration and are in the process of being installed for real-world use in the museums. It has been shown that the multi-scale colour coherence vector (M-CCV) technique can provide effective sub-image retrieval and is also applicable when the sub-mage and target are captured at different resolutions, the wavelet transform has formed the basis for a facility for query by fax and by other low quality images, and work has been presented on query by crack type.

The work of the Artiste project is being continued under a further European Project, Sculpteur, in which the database will be extended to hold 3D models of museum artefacts and the system will be integrated with emerging semantic web technology to enrich the knowledge base on the collections from the museums and galleries.

Acknowledgment

The authors are grateful to their collaborators on the ARTISTE project: NCR (Dk), Giunti Interactive Labs (I), Uffizi Gallery (I), IT Innovation Centre (UK), The National Gallery (UK). We also thank Dr Spike Bucklow (Hamilton-Kerr Institute Cambridge) for collaboration on craquelure types.

References

[1] W. Vaughan, “Automated Picture referencing; Another look at Morelli,” *Computers and the History of Art,* Harwood Academic Publishers, vol. 2, no. 2, 1991, pp.7-18.

[2] B. Holt and K. Weiss, “The QBIC Project in the Department of Art and Art History at UC Davis”. In *The 60th ASIS Annual Meeting* 1997, Digital Collections: Implications for users, funders, developers and maintainers, vol. 34, ASIS, 1997, pp 189-195.

[3] A. Del-Bimbo, M. Mugnaini, P. Pala and F. Turco, “Visual querying by color perceptive regions”, Pattern Recognition, vol. 31, no. 9, 1998, pp 1241-1253.

[4] RDF Resource Description Framework http://www.w3.org/RDF/

[5] DublinCore metadata initiative http://www.dublincore.org/

[6] J. Cupitt and K. Martinez, “VIPS: an image processing system for large images”, *Proc. SPIE conference on Imaging Science and Technology, San Jose*, vol. 2663, 1996, pp 19-28.

[7] P. Allen and M. Boniface and P. Lewis and K. Martinez, “Interoperability between Multimedia Collections for Content and Metadata-Based Searching”, *Online Proceedings of the World Wide Web conference*, Hawaii 2002, available: http://www2002.org/CDROM/alternate/196/index.html.

[8] MySQL: http://www.mysql.com.

[9] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, “Content-Based Image Retrieval at the End of the Early Years,” *IEEE Trans. On Pattern Analysis And Machine Intelligence,* vol. 22, pp, December 2000, 1349-1380

[10] M. J. Swain and D. H. Ballard. "Color Indexing ", *International Journal of Computer Vision,* vol. 7, no. 1, 1991. pp 11-32

[11] G. Pass, R. Zabih, and J. Miller. "Comparing Images Using Color Coherence Vectors", Proceedings of *ACM Multimedia*, 1996. pp 65-73

[12] M. Westmacott, P. Lewis, K. Martinez, "Using colour pair patches for image retrieval", *Proceedings of the IS&T First European Conference on Colour in Graphics, Imaging and Vision*, April 2002, pp. 245-247.

[13] S. Chan, K. Martinez, P. Lewis, C. Lahanier and J. Stevenson, "Handling Sub-Image Queries in Content-Based Retrieval of High Resolution Art Images", *Proc. International Cultural Heritage Informatics Meeting,* vol. 2, Sept. 2001, pp.157-163.

[14] S. G. Mallat, "A theory for multiresolution signal decomposition: The Wavelet Representation," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* Vol.11, July 1989 pp. 674-693

[15] Wavelab Toolbox ver 802 for Matlab, available: http://www-stat.stanford.edu/wavelab/

[16] F. S. Abas and K. Martinez, "Classification of painting cracks for content-based analysis", *Proc. SPIE conference on Machine Vision Applications in Industrial Inspection XI*, Jan 2003.

[17] S. Bucklow, "A Stylometric Analysis of Craquelure", *Computers and Humanities,* vol. 31, 1989, pp. 503-521.

[18] F. Meyer, "Iterative Image Transformations for an Automatic Screening of Cervical Smears", *J.Histoch. Cytochem*, vol. 27, 1979.

[19] I. Giakoumis, I. Pitas, "Digital Restoration of Painting Cracks", in *ISCAS '98, Proceedings of the IEEE International Symposium on Circuits and Signals*, 31 May-3, June 1998, pp. 269-272.

[20] N. Otsu, "A Threshold Selection Method From Gray-level Histogram", *IEEE Trans. on Systems, Man, Cybernetics*, vol. 9, no. 1, 1979, pp. 62-69.

[21] H. Freeman, "Boundary Encoding and Processing", *Picture Processing and Psychopictories*, Academic Press, 1970.

[22] F. S. Abas and K. Martinez, "Craquelure Analysis for Content-Based Retrieval", *Proc. IEEE14th International Conference on Digital Signal Processing*, July 2002, pp.111-114.

LIST OF FIGURES

Fig. 1 The image of the baby represents a query, and the result of sub-image matching is shown as a white boundary in the *Moise présenté à Pharaon* image on the right.

Fig. 2. The query image and the top three matches (distances 5032, 10784, 11754).

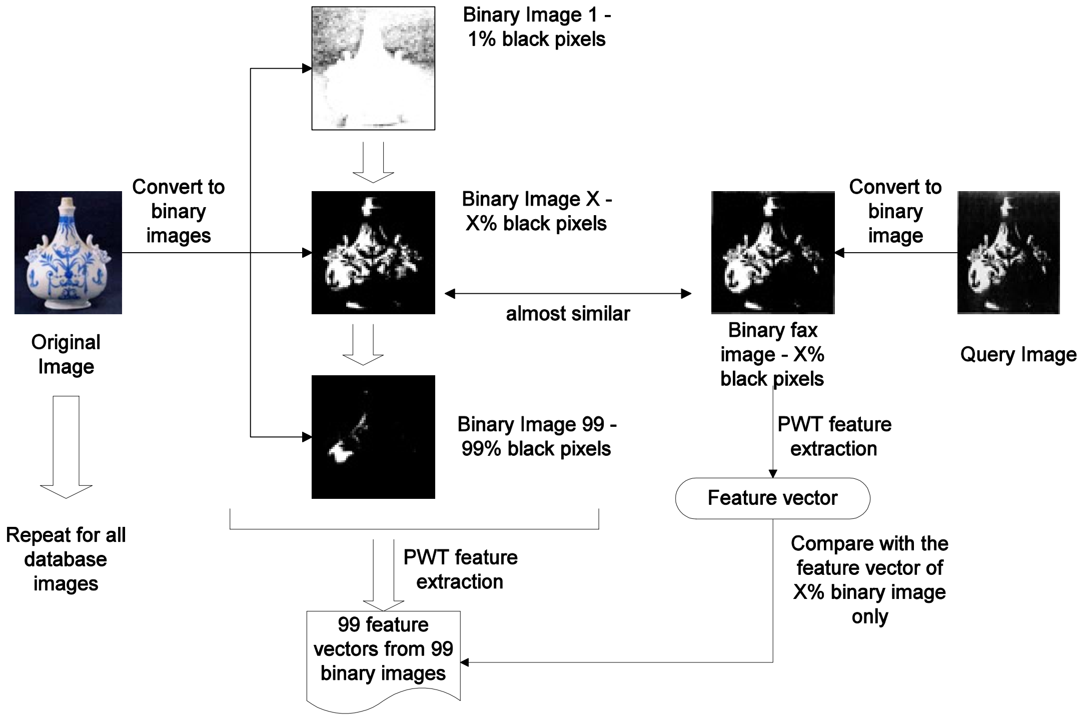


Figure 3 Binary image matching between fax image and its original

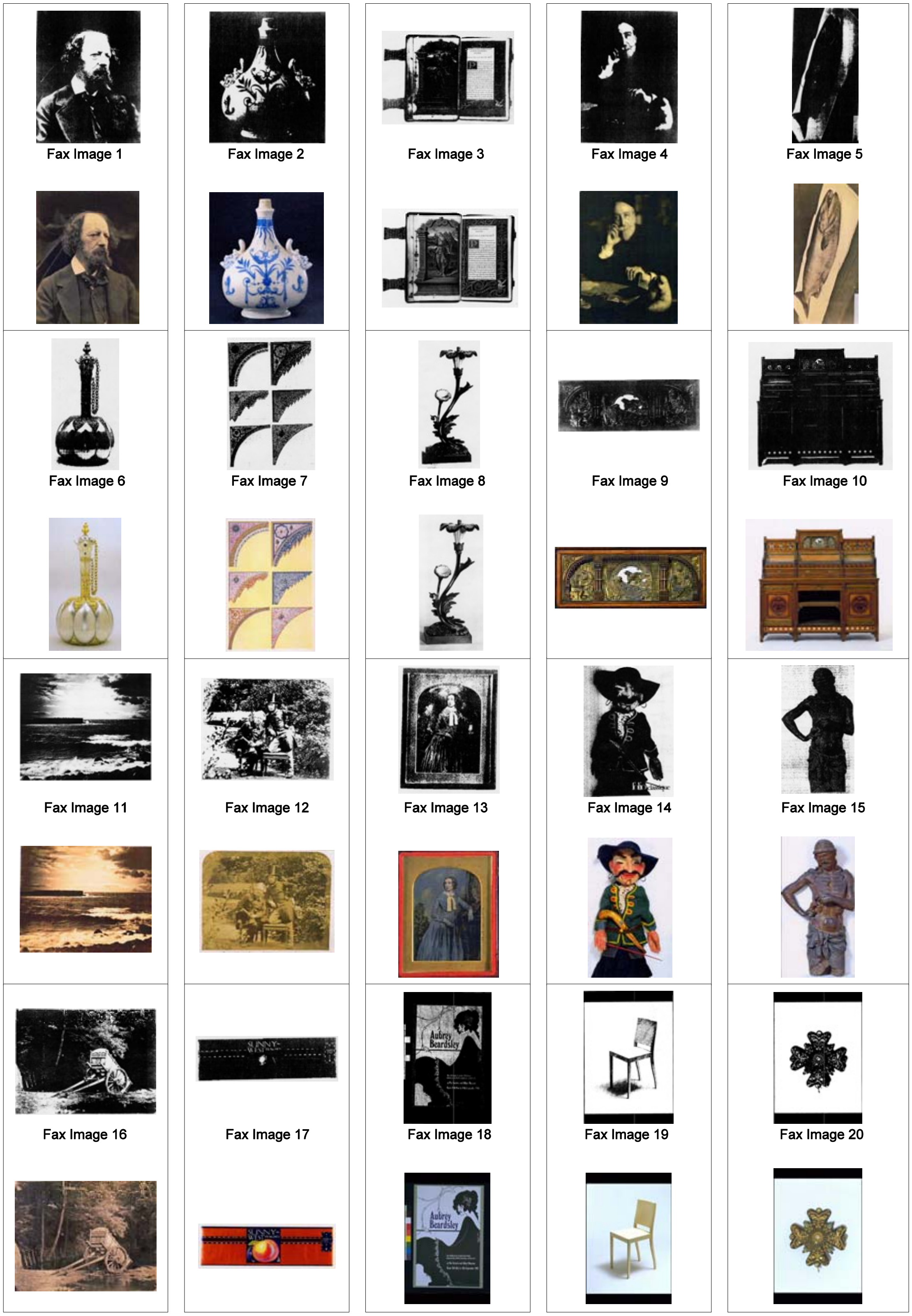


Figure 4 The fax images used in this experiment and their originals.

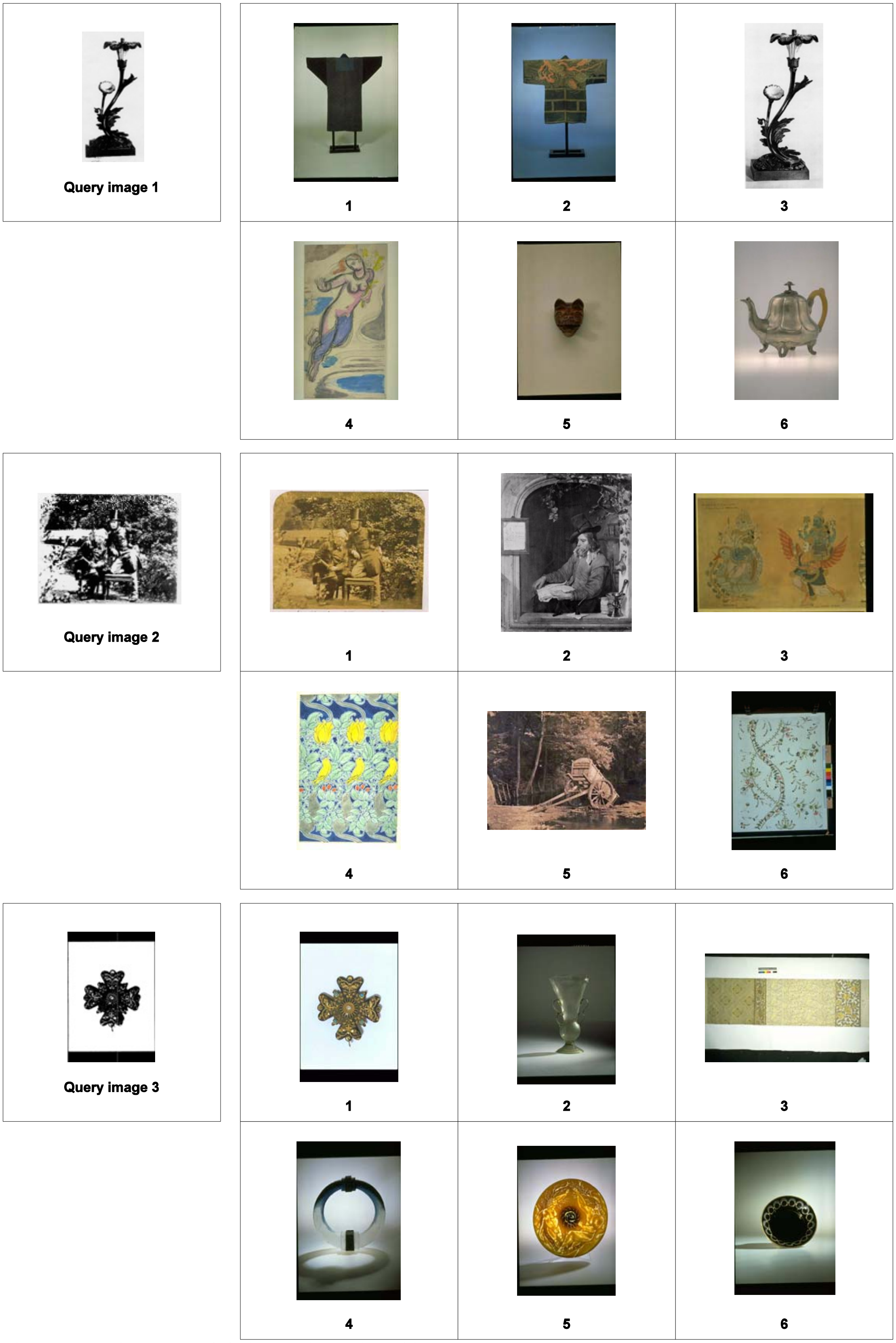


Figure 5 Fax images and their top six retrieved images

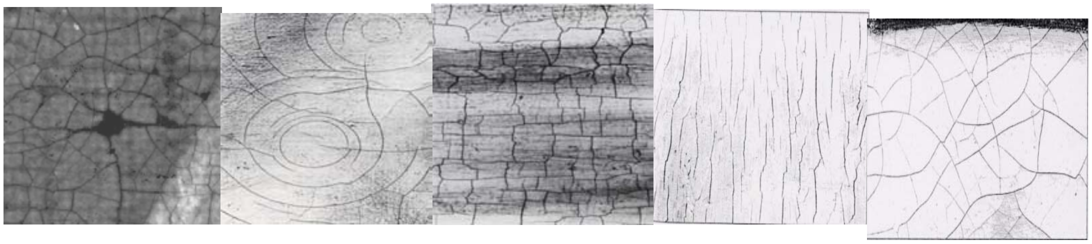


Figure 6 (from left to right, top to bottom)) Crack classes: spider-web, circular, rectangular, unidirectional, random.

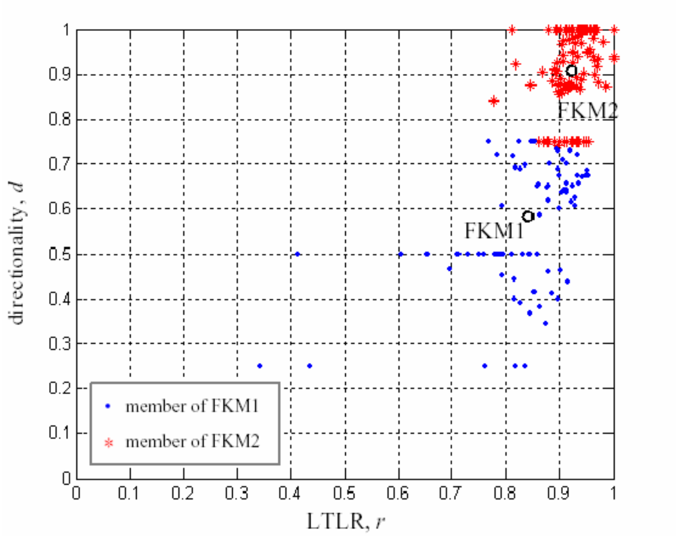


Figure 7: Results of fuzzy K-means clustering of two crack features. FKM1,2 show cluster centres.