Content-based image retrieval using a mobile device as a novel interface

Jonathon S. Hare and Paul H. Lewis

Intelligence, Agents, Multimedia Group,
School of Electronics and Computer Science,
University of Southampton, Southampton, SO17 1BJ, United Kingdom

ABSTRACT

This paper presents an investigation into the use of a mobile device as a novel interface to a content-based image retrieval system. The initial development has been based on the concept of using the mobile device in an art gallery for mining data about the exhibits, although a number of other applications are envisaged. The paper presents a novel methodology for performing content-based image retrieval and object recognition from query images that have been degraded by noise and subjected to transformations through the imaging system. The methodology uses techniques inspired from the information retrieval community in order to aid efficient indexing and retrieval. In particular, a vector-space model is used in the efficient indexing of each image, and a two-stage pruning/ranking procedure is used to determine the correct matching image. The retrieval algorithm is shown to outperform a number of existing algorithms when used with query images from the mobile device.

Keywords: Content-based Image Retrieval, Salient regions, Mobile Device, Vector-space, Local Descriptors, tf-idf

1. INTRODUCTION

Given the large amount of research into content-based image retrieval currently taking place, new interfaces to systems that perform queries based on image content need to be considered. In this paper a new paradigm for content-based image retrieval is introduced, in which a mobile device is used to capture the query image and display the results. The system consists of a client-server architecture in which query images are captured on a mobile device and then transferred to a server for further processing. The server then returns the results of the query to the mobile device. There are a number of possible user-scenarios for the use of such a device. These scenarios generally fall into two categories, depending on what kind of query result the system would be expected to provide.

The first category is very much like previous research on the “physical hyper-link” carried out at HP labs,[1] where a user can ‘click’ on real world objects as if they were a hyper-link, using a mobile device as the interface. In this case, the objective of the system is to find an *exact* representation of the query image in the database and to return metadata corresponding to the object represented in the query image. For example, consider using the device in a museum or art gallery. The device could be pointed at various exhibits or paintings and would return metadata about that particular object. Another possible example would be in a bookshop. In this case the device could be pointed at a book cover, and the returned metadata could be, for example, reviews of that particular book.

The second category is much more like classical content-based image retrieval. In this case, the objective is not necessarily to find an exact match, but rather to find a ranked set of *similar* images - either visually similar (e.g. in terms of colour) or similar in terms of the semantics of the content.

This paper examines the first category in detail, although the retrieval algorithms presented are equally applicable to the second category. The paper is split into several sections. The first section presents our novel content-based retrieval model. The second section shows how the retrieval model has been implemented in a client-server architecture. The third section illustrates some results of our system in a mock museum scenario. Finally, the last section wraps up with some conclusions and presents some ideas for future research.

Further author information: E-mail: jsh02r@ecs.soton.ac.uk, Telephone: +44 (0)2380 595415

2. CONTENT-BASED IMAGE RETRIEVAL

The processing performed by the server part of our system is based around a novel content-based image retrieval algorithm that uses local image descriptors to describe image content and an indexing methodology that is inspired by techniques from the information retrieval community. The retrieval algorithm exhibits a number of features that make it stand apart from some of the existing algorithms. The most important of these is the ability of the algorithm to retrieve a matching image from the database given a query image that is heavily degraded by noise and has been transformed by a planar homography. The algorithm is also remarkably resistant to scale changes between the query image and database images.

2.1. Salient regions for content-based image retrieval

Much previous work in the field of content-based retrieval has been based around the concepts of using global descriptors to describe the content of the image. More recently researchers have begun to realise that global descriptors are not necessarily good when it comes to describing the actual objects within the images and their associated semantics. Two approaches have grown from this realisation; firstly approaches have been developed whereby the image is segmented into multiple regions, and separate descriptors are built for each region; and secondly, the use of salient points has been suggested.

The first approach has been demonstrated to work,[2] although it has a large problem - that of how to perform the segmentation. Over the years many techniques for performing image segmentation have been suggested, although none really solve the problem of linking the segmented region to the actual object that is being described. Indeed, this shows that the non-naive segmentation problem is not just a bottom-up image processing problem, but also a top-down problem that requires knowledge of the true object, before it can be successfully segmented.

The second approach avoids the problem of segmentation altogether by choosing to describe the image and its contents in an altogether different way. By using salient points or regions within an image, it is possible to derive a compact image description based around the local attributes of the salient points. A number of different methods for finding salient points have been suggested, from the simple Harris & Stephens[3] corner detector, to wavelet based approaches,[4–6] to methods centred around image entropy.[7, 8] Many previous approaches to using salient points have generated feature-vectors from pixel data in fixed-sized regions around the salient point, usually a 3x3 or 9x9 pixel neighbourhood centred on the point,[5] although some of the modern state-of-the-art detectors find affine invariant regions and generate descriptors from within the region.[9–11] In previous work, it was shown that content-based retrieval based on salient interest points and regions performs much better than global image descriptors.[5, 12] For our content-based image retrieval algorithm, we select salient regions using the method described by Lowe,[13] where scale-space peaks are detected in a multi-scale difference-of-Gaussian pyramid. Peaks in a difference-of-Gaussian pyramid have been shown to provide the most stable interest regions when compared to a range of other interest point detectors.[12, 14] Figure 1 illustrates the kind of regions found from peaks in a difference-of-Gaussian pyramid. Local feature descriptors are generated for each of these salient regions.

2.2. Local Feature Descriptors

There are a large number of different types of feature descriptors that have been suggested for describing the local image content within a salient region; For example colour moments and Gabor texture descriptors.[5] However, many of these descriptors are not robust to the imaging conditions. Unfortunately, the quality of images captured with the mobile device is fairly poor. The captured images exhibit a grainy noise pattern, poor contrast, poor colour quality and blurring (both from motion, as the camera takes a reasonable amount of time to grab a shot, during which the user must hold the device stationary, and also due to the manual focus control of the camera). A typical query image captured by the device is shown in Figure 2. These problems make many of the traditional local feature descriptors fairly useless in matching between a database of high quality images and query images from the mobile device. An alternative is to use a feature descriptor that performs well in the presence of poor imaging conditions. In the current implementation of the system, Lowe's SIFT (Scale Invariant Feature Transform) descriptor[13] is used. The SIFT descriptor was shown to be superior to other descriptors found in the literature,[15] such as the response of steerable filters or orthogonal filters. The performance of the

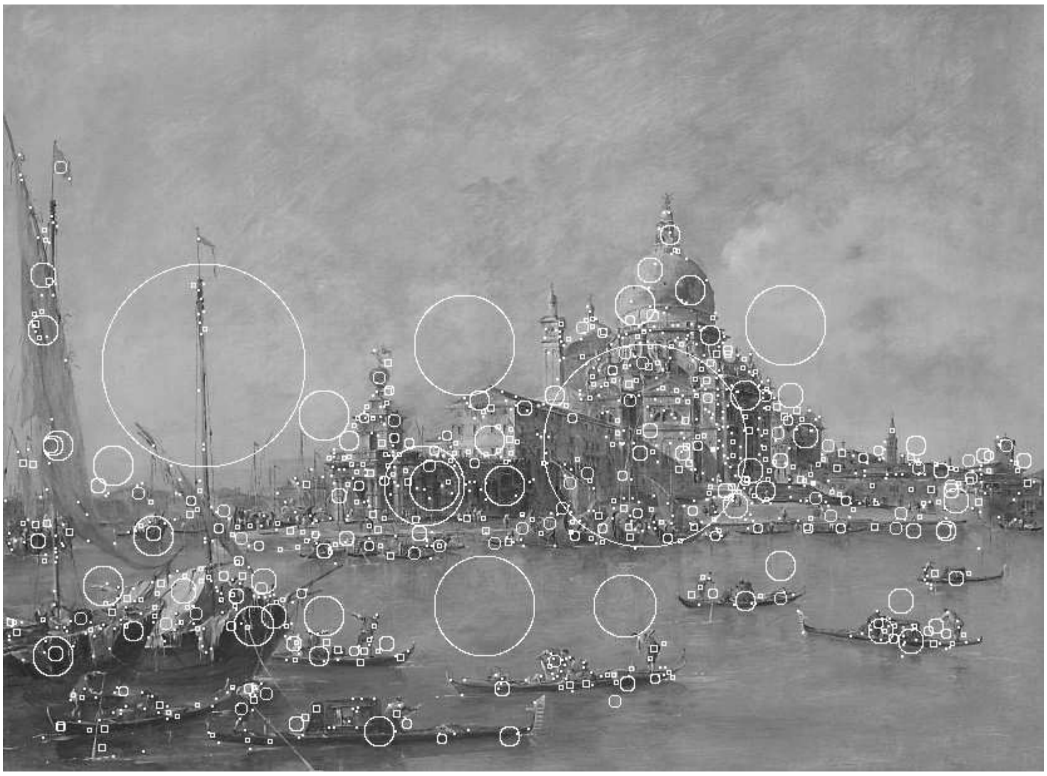


Figure 1. Example image showing salient regions found from peaks in the difference-of-Gaussian pyramid. Image Copyright © 2004, National Gallery, London, All rights reserved.

SIFT descriptor is enhanced because it was designed to be invariant to small shifts in the position of the salient region, as might happen in the presence of imaging noise.

2.3. Indexing and retrieval

Our original approach to retrieval based on images described by salient regions worked by comparing each individual salient region in a query image with all the salient regions of each image in the database.[12] This is a very expensive operation, especially if the database is large. For example a typical query image may have around 500 salient regions, and each image in the database may have anything between 1,000 and 5,000 regions approximately. With a database of around 1,000 images, this means that each salient region in the query image needs to be compared to between 1,000,000 and 5,000,000 other regions. This is itself an expensive operation as the SIFT feature vectors are themselves made up of 128 dimensions.

In this paper, we describe a different method based on a classical approach to retrieval from the information retrieval field. Recent work by Sivic and Zisserman[10] and slightly earlier work by Westmacott and Lewis,[16] showed a new approach to object matching within images and video footage. The approach was based on an analogy with classical text retrieval using a vector-space model. Instead of comparing individual salient regions, a descriptor is built for each image in the database that describes the whole content. This descriptor has much lower dimensionality than the combined local descriptors of each of the salient regions and enables significant reductions in computational load, and thus enables much faster searching of the database.

2.3.1. The Vector-Space Model of Classical Text Retrieval: A Brief Overview

Most classical text retrieval systems work in the same general way, by representing a document and query as a set of terms. These terms are represented as axes in a vector-space, using weighted term frequency as the

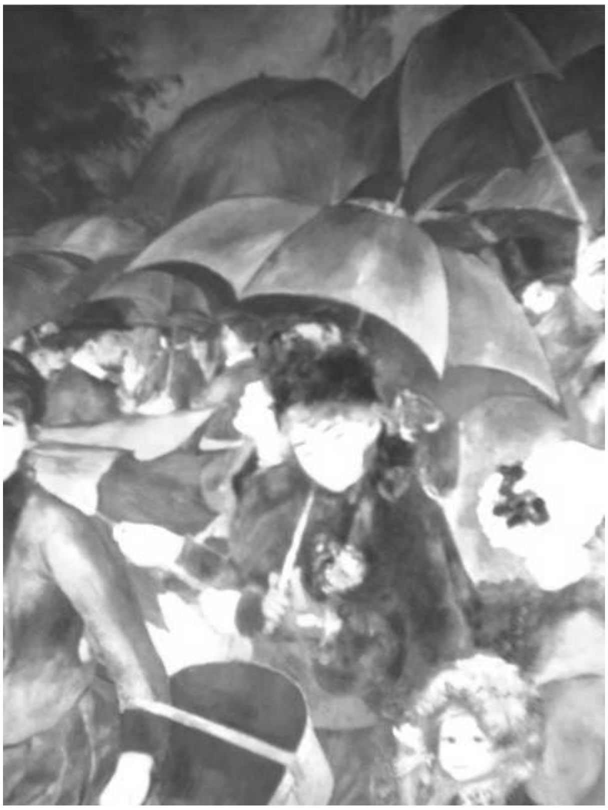


Figure 2. A typical query image, illustrating the poor imaging conditions (motion blur, blurring, noise, contrast, etc.). Image Copyright © 2004, National Gallery, London, All rights reserved.

distance along the axis corresponding to that term. Described below are a number of standard steps for this model.

Parsing and Stemming: Firstly, a document is parsed into a list of separate words, this is obviously an easy task in most languages as the words are separated by spaces. The words are then transformed by a process called stemming. The stemming process represents words by their stems, for example, 'CONNECT', 'CONNECTED', 'CONNECTING', 'CONNECTION', and 'CONNECTIONS' are all represented by the stem 'CONNECT'. Words with a common stem will often have similar meanings. Various algorithms for stemming have been developed, for example, the Porter Stemmer,[17] that stems English words.

Stop Lists: The next stage is to apply a stop list. The stop list is used to reject common words that occur frequently throughout the corpus of documents, and therefore are not discriminating for a particular document. Examples of such words include words like 'and', 'an' and 'the'.

Representing documents by word frequency: Each of the words from the document (after application of the stop list) are then represented by a unique identifier for that word. The number of occurrences of each word in the document is then counted and a vector of word-frequencies is created to represent the document.

Frequency weighting: Each component of the vector of word frequencies is often weighted. In the case of the Google web search engine, the weighing of a particular web page depends on the number of pages linking to that particular page.[18]

The standard way of weighting the frequency vectors of text documents is called 'term frequency-inverse document frequency', *tf-idf*, and is computed as follows. Suppose that there is a vocabulary of k words, then each document is represented by a k-vector $V_d = (t_1, \ldots, t_i, \ldots, t_k)^T$, of weighted word frequencies with components,

$$t_i = \frac{n_{id}}{n_d} \log \frac{N}{n_i} \tag{1}$$

where n_{id} is the number of occurrences of word i in document d, n_d is the total number of words in the document d, n_i is the number of occurrences of the term i in the whole database and N is the number of documents in the whole database. The weighting is the product of two terms: the *word frequency* n_{id}/n_d and the *inverse document frequency* $\log N/n_i$. The intuition is that word frequency increases the weights of words that occur frequently in a particular document, and thus describe it well, whilst the inverse document frequency down-weights words that appear often in the database.

Indexing using Inverted Files: Inverted file structures are used for efficient retrieval. An inverted file is like an ideal book index. Each word in the collection has an entry in the inverted file, together with a list of documents (and the position in which the word occurs in the document) that contain that word.

Searching: Ranking the results: In order to search the database of documents, a *tf-idf* vector is created for the query terms or document, and the query vector is compared against all the vectors in the database, V_d. The documents in the database are ranked using the normalised scalar product (cosine of angle):

$$\cos(\theta) = \frac{\mathbf{A} \bullet \mathbf{B}}{|\mathbf{A}||\mathbf{B}|} \tag{2}$$

2.4. Applying the vector-space model to image retrieval

In this section, the ideas and methods described above for text retrieval are taken and applied to image retrieval. The analogy used is that an image is a document, and consists of multiple terms, or 'visual' words. In the previous sections, the use of saliency as a means to build image descriptions was discussed. In order to build the 'visual' words for an image, it is suggested that each 'visual' word is formed from the local description of the image in a salient region.

2.4.1. Building 'visual' words: Vector Quantisation

One immediately obvious problem with taking local descriptors to represent words is that, depending on the descriptor, there is a possibility that two very similar image patches will have slightly different descriptors, and thus there is a possibility of having an absolutely massive vocabulary of words to describe the image. A standard way to get around this problem is to apply vector quantisation to the descriptors to quantise them into a known set of descriptors. This known set of descriptors then forms the vocabulary of 'visual' words that describes the image. This process is essentially the equivalent of the stemming, where the vocabulary consists of all the possible stems. The next problem is that of how to design a vector quantiser. Sivic and Zisserman[10] selected a set of video frames from which to train their vector quantiser, and used the k-means clustering algorithm to find clusters of local descriptors within the training set of frames. The centroids of these clusters then became the 'visual' words representing the entire possible vocabulary. The vector quantiser then proceeded by assigning local descriptors to the closest cluster.

In this work, a similar approach was used. A sample set of images from the data-set was chosen at random, and feature vectors were generated about each salient region in all the training images. Clustering of these feature descriptors was then performed using the batch k-means clustering algorithm with random start points in order to build a vocabulary of 'visual' words. Each image in the entire data-set then had its feature vectors quantised by assigning the feature vector to the closest cluster.

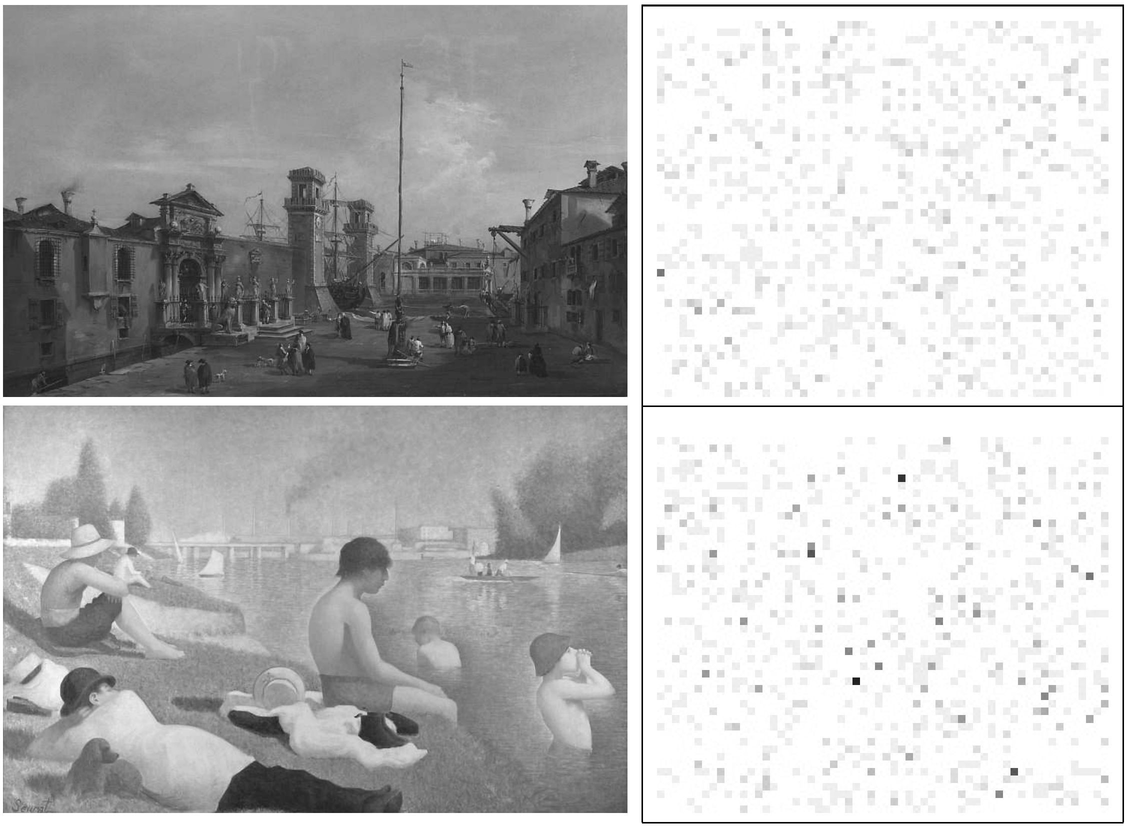


Figure 3. Illustration of the word-frequency vectors of two images with a 3000 word vocabulary. The vectors have been projected onto a 60x50 array. Images Copyright © 2004, National Gallery, London, All rights reserved.

2.4.2. Image Retrieval based on 'visual' words

Given the lists of 'visual' words for each image in the data-set, the next stage is to calculate a word-frequency vector to represent each image. Figure 3 illustrates the word-frequency vectors, using a 3000 word vocabulary, of two images by showing the histogram of word frequencies projected onto a 60x50 array, with the darker colours representing higher values. The figure clearly shows that the distribution of frequencies of 'visual' words for each of the images are different. The *tf-idf* weighting (Equation 1) is used to weight each element of the word-frequency vector. In order to perform actual retrieval, a query vector, V_q, is constructed from the query image, and all the documents in the database are ranked by the normalised scalar product (Equation 2) between the query vector V_q and the each document vector V_d.

Getting the correct match: Due to the way the indexing scheme works the top ranking matching image may not actually be a representation of the query image. This is due in part to the imaging conditions, but also to the fact that the query image is likely to be either a sub-image or super-image of the matching representation in the database. In order to find the actual matching image, we re-rank the top N results based on the geometric consistency of the salient regions. This is akin to text-search methodologies where the rank of the document is increased if the query terms occur with similar positional relations to each other in both the query and document.

Because the aim of the system is to recognise planar objects, we model the geometric consistency of the salient regions as a planar homography. In order to perform the re-ranking, we test each of the top N ranked images' salient regions for a consistent homography between the query image's salient regions using the RANSAC algorithm to robustly ascertain whether a consistent homography exists.[19]

2.5. Summary

In summary, we have presented an image retrieval methodology with a two-stage re-ranking procedure. The algorithm transforms the query image into a vector-space based on the frequencies of ‘visual’ words within the image. The ‘visual’ words are created in such a way as to be invariant to a range of transformations, including changes in homography, intensity changes and imaging noise. The first stage ranking procedure uses the cosine similarity of weighted ‘visual’ word frequency vectors to rank the images in the database. The second stage re-ranks the top N results based on the geometric consistency between the salient regions of the query and N results. The outcome is that the highest ranked image should correspond to the query. The overall retrieval process is illustrated in Figure 4.

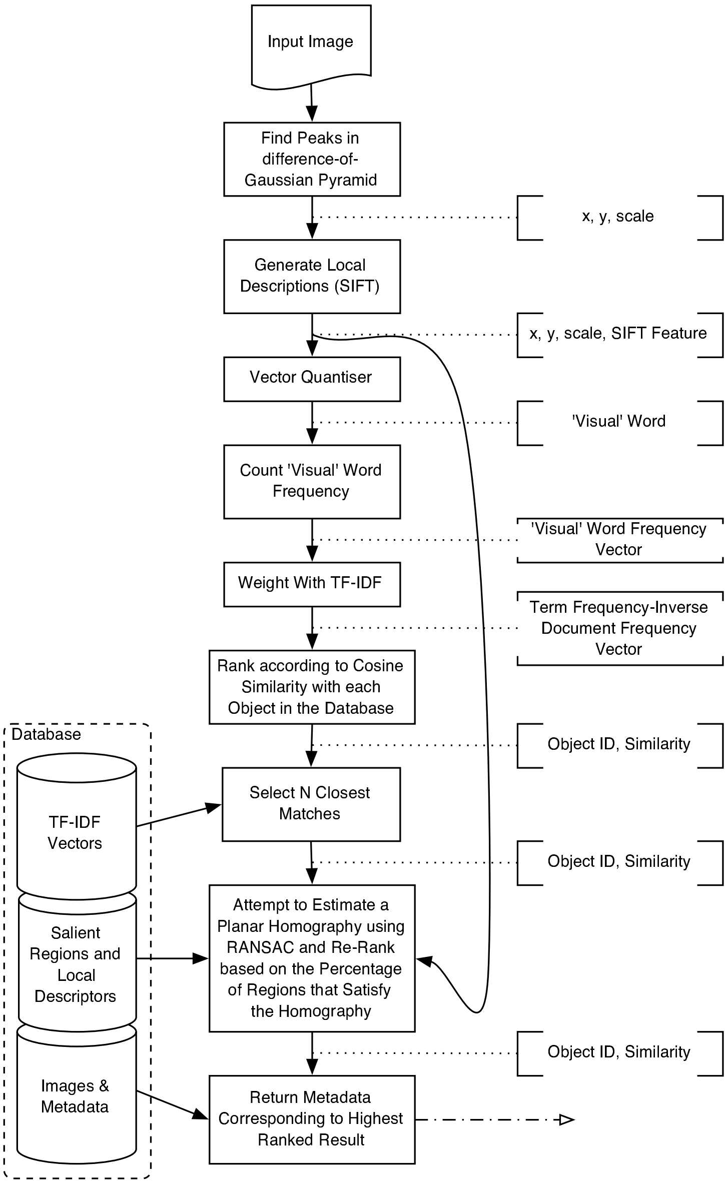

Input Image
Find Peaks in difference-of-Gaussian Pyramid
x, y, scale
Generate Local Descriptions (SIFT)
x, y, scale, SIFT Feature
Vector Quantiser
'Visual' Word
Count 'Visual' Word Frequency
'Visual' Word Frequency Vector
Weight With TF-IDF
Term Frequency-Inverse Document Frequency Vector
Rank according to Cosine Similarity with each Object in the Database
Object ID, Similarity
Database
TF-IDF Vectors
Select N Closest Matches
Object ID, Similarity
Salient Regions and Local Descriptors
Attempt to Estimate a Planar Homography using RANSAC and Re-Rank based on the Percentage of Regions that Satisfy the Homography
Object ID, Similarity
Images & Metadata
Return Metadata Corresponding to Highest Ranked Result

Figure 4. Overview of our content-based image retrieval technique.

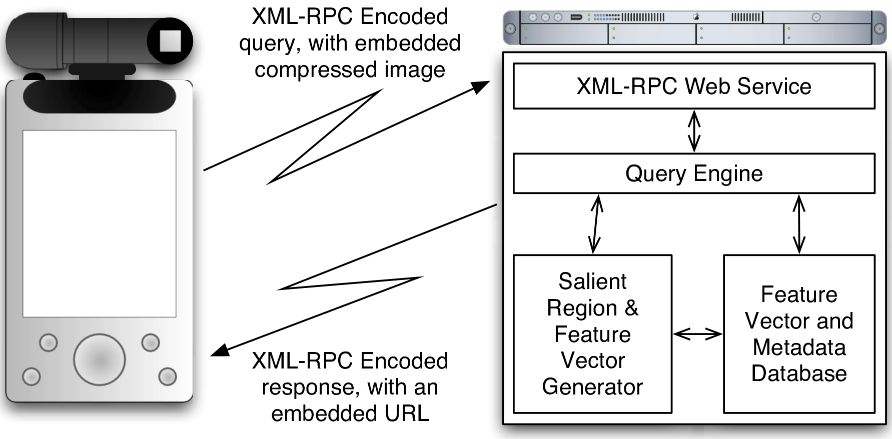

XML-RPC Encoded query, with embedded compressed image
XML-RPC Web Service
Query Engine
Salient Region & Feature Vector Generator
Feature Vector and Metadata Database
XML-RPC Encoded response, with an embedded URL

Figure 5. An overview of the mobile image retrieval system.

3. CLIENT-SERVER IMPLEMENTATION AND TECHNOLOGY

In order to develop our mobile architecture for retrieval, a test-bed has been constructed from commercially available equipment, and using open standards for data transfer. The current implementation of the system consists of a mobile device with a camera (an HP h5550 iPAQ Pocket PC and Lifeview FlyCAM SD) acting as a mobile client, and a PC acting as a server. The mobile client is connected to the Internet through a wireless connection (either Bluetooth or 802.11b). The server machine hosts a web service to which the client can connect and transmit JPEG compressed query images. XML remote procedure calls (XML-RPC) are used to provide the interface to the server. The server processes the queries it receives and returns the result to the client. Figure 5 illustrates the topology of the system.

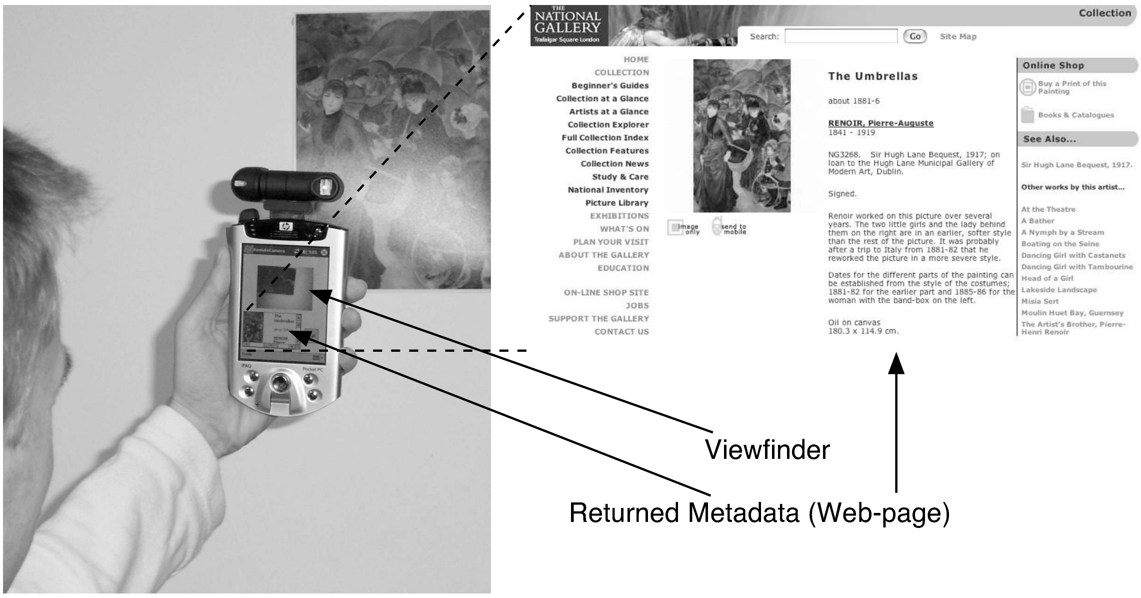
Figure 6 illustrates the use of the device in an art gallery scenario. The server has been configured to return a web-page with information corresponding to the database image that most closely matches the query. The web-page is then displayed on the client.

4. RETRIEVAL PERFORMANCE

The performance of the retrieval algorithm was evaluated by testing 200 randomly selected images captured using the mobile device and looking at the rank of the matching image in the returned set. Obviously, the ideal scenario is that the matching image is always returned in the highest ranking (rank 0) position. The image database consisted of over 850 images from the National Gallery image collection. A number of sample query images are shown in Figure 7.

Figure 8 illustrates the effect of querying the database with a number of different retrieval algorithms, including the Colour Coherence Vector (CCV) algorithm,[20] RGB Colour Histogram matching, Grey-level Histogram matching, Pyramid-structured Wavelet Transform (PWT) algorithm,[21] and the vector-space retrieval algorithm detailed in the previous sections, without the second-stage re-ranking. The graph shows that the vector-space retrieval algorithm performs dramatically better than the other algorithms; in fact, the performance of the other algorithms is little better than randomly choosing an image from the database. Just under 35% of matching images using the vector-space algorithm were found in the highest ranking position, and the percentage of matched images drops off rapidly as rank increases.

The effect of the second-stage re-ranking was also investigated. The purpose of the two-stage re-ranking approach is to reduce computational load. The first retrieval stage identifies possible matches, and the second-stage verifies the actual match. If the second stage re-ranking were performed on all the images in the database, the probability of identifying a correct match is extremely high, but the computational load would be massive



The Umbrellas

about 1881-6

RENOIR, Pierre-Auguste
1841 - 1919

NG3268. Sir Hugh Lane Bequest, 1917; on loan to the Hugh Lane Municipal Gallery of Modern Art, Dublin.

Signed.

Renoir worked on this picture over several years. The two little girls and the lady behind them on the right are in an earlier, softer style than the rest of the picture. It was probably after a trip to Italy from 1881-82 that he reworked the picture in a more severe style.

Dates for the different parts of the painting can be established from the style of the costumes; 1881-82 for the earlier part and 1885-86 for the woman with the band-box on the left.

Oil on canvas
180.3 x 114.9 cm.

Figure 6. The system in use in a mock art gallery scenario. Images Copyright © 2004, National Gallery, London, All rights reserved.

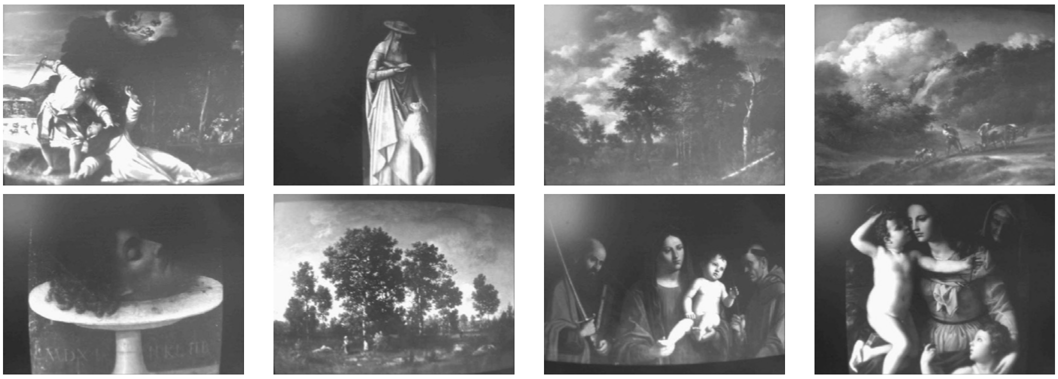


Figure 7. Example query images captured by the mobile device for testing the performance. Images Copyright © 2004, National Gallery, London, All rights reserved.

and the need for the first-stage retrieval would be negated. By considering only the top N ranking matches from the first-stage in the second-stage, computational load is dramatically reduced at the expense of retrieval performance. Figure 9 illustrates the effect of changing N versus the rate of correct retrieval, where correct retrieval is defined as the image matching the query being in the highest ranking position after the second-stage re-ranking. The graph shows that a first-place recognition rate in excess of 80% can be achieved by performing the geometry based re-ranking procedure on the top 20 ranked matches from the first-stage retrieval.

4.1. Discussion

The results presented above were found using a naive set of parameters for things such as the number of 'visual' words in the vocabulary. It is possible that by tuning the parameters, the retrieval performance could be further improved. The vector quantiser used for the experiments was certainly non-optimal for the test image data-set, and no investigation into the optimal number of 'visual' words in the vocabulary was performed. Performance

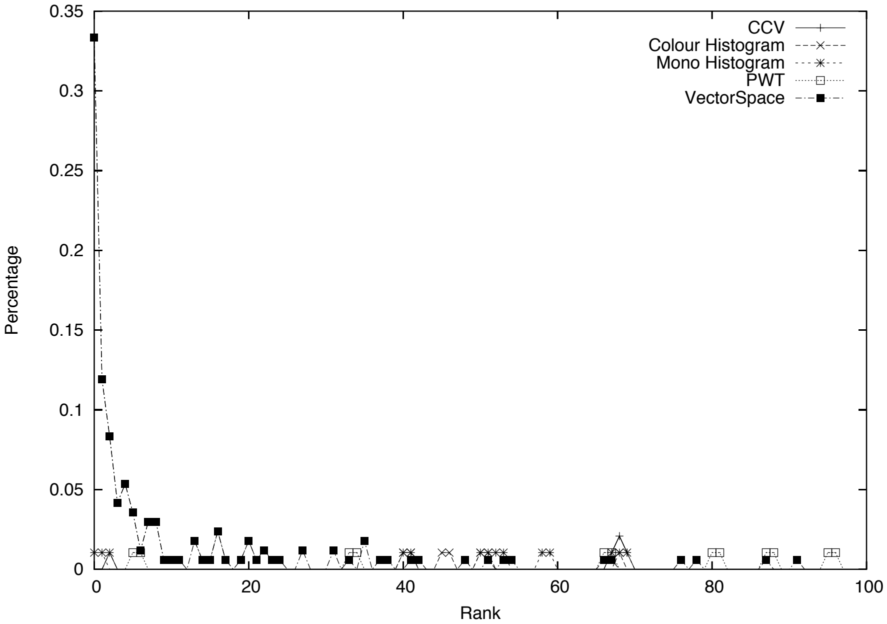


Figure 8. Plot of the rank of the matching image for a number of different retrieval algorithms.

could also possibly be improved by pre-processing the query images to remove the radial lens distortion the camera exhibits and also by normalising the images. However, despite these non-optimised parameters, the results show that the two-stage retrieval algorithm performs well when presented with query images of low quality, such as those from a mobile device.

5. CONCLUSIONS AND FUTURE WORK

This paper has presented a novel two-stage approach to image retrieval that has been designed to work especially well with queries that have been severely degraded by the imaging apparatus. The retrieval algorithm is able to quickly narrow-down a set of likely candidate matching images and then re-rank these to find the most likely match. A system which uses the retrieval algorithm in a client-server manner with a mobile device as the client interface has been presented, and the results look promising. The results have shown that the new algorithm performs significantly better than existing retrieval algorithms, which have a retrieval performance only slightly better than choosing results at random from the database, when used with the degraded query images.

A number of items need further investigation. The vocabulary of 'visual' terms used in the experiments is certainly non-optimal for the image database used. Optimising this vocabulary should improve the performance of the first-stage of the algorithm, thus improving recognition rates for an equivalently sized N in the second-stage. The issue of how the vocabulary should be generated also needs to be examined, as k-means is probably not the most optimal approach.

The system does not at present use any colour information for the retrieval. We plan to address this by augmenting the SIFT descriptors with a local colour descriptor. Again, this should help improve retrieval performance. Finally, we also want to investigate the use of a list of "stop-words" in the vector-space algorithm to ignore commonly occurring 'visual' words. We also wish to implement an inverted-indexing scheme to improve the speed of retrieval.

ACKNOWLEDGMENTS

We are grateful to the Engineering and Physical Sciences Research Council and the Motorola UK Research Laboratory for their support of this work. We would also like to thank the National Gallery for providing the

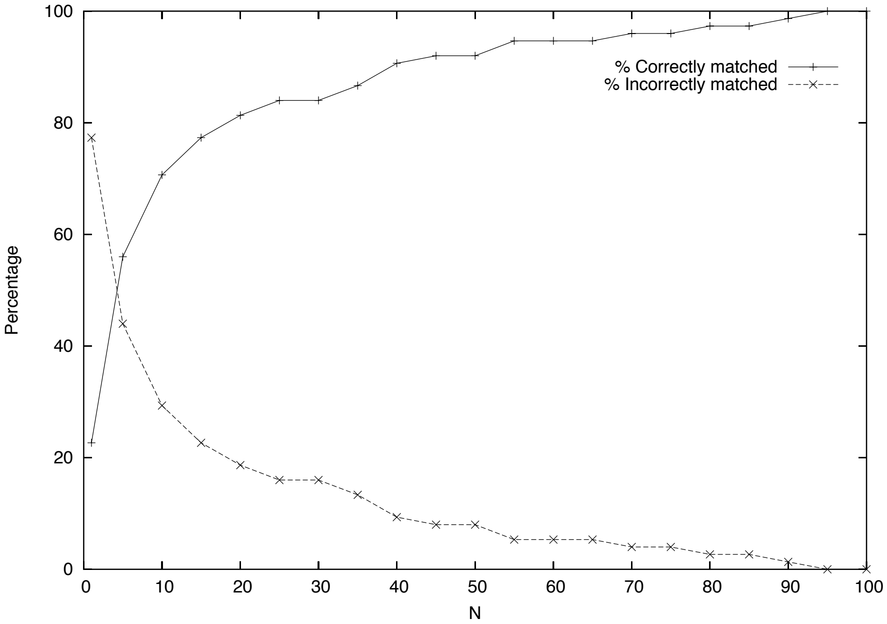


Figure 9. Retrieval rate versus N, the number of images considered for second-stage geometry based re-ranking.

image data-set used for testing the system.

REFERENCES

1. J. Barton and T. Kindberg, "The challenges and opportunities of integrating the physical world and networked systems," Tech. Rep. HPL-2001-18, HP Labs, 2001.
2. C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld: Image segmentation using expectation-maximization and its application to image querying," *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(8), pp. 1026–1038, 2002.
3. C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the 4th ALVEY vision conference*, M. M. Mathews, ed., pp. 147–151, (University of Manchester, England), 1988.
4. A. Shokoufandeh, I. Marsic, and S. Dickinson, "View-based object recognition using saliency maps," *Image Vis. Comput.* **17**(5-6), pp. 445–460, 1999.
5. N. Sebe, Q. Tian, E. Loupias, M. Lew, and T. Huang, "Evaluation of salient point techniques," *Image and Vision Computing* **21**, pp. 1087–1095, 2003.
6. N. Sebe and M. S. Lew, "Comparing salient point detectors," *Pattern Recognition Letters* **24**(1-3), pp. 89–96, 2003.
7. T. Kadir, *Scale, Saliency and Scene Description.* PhD thesis, University of Oxford, Department of Engineering Science, Robotics Research Group, University of Oxford, Oxford, UK, 2001.
8. T. Kadir and M. Brady, "Saliency, scale and image description," *Int. J. Comput. Vis.* **45**(2), pp. 83–105, 2001.
9. T. Tuytelaars and L. V. Gool, "Content-based image retrieval based on local affinely invariant regions," in *Third International Conference on Visual Information Systems*, pp. 493–500, 1999.
10. J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *International Conference on Computer Vision*, pp. 1470–1477, October 2003.
11. S. Obdrzalek and J. Matas, "Image retrieval using local compact DCT-based representation," in *DAGM-Symposium 2003*, pp. 490–497, 2003.

12. J. S. Hare and P. H. Lewis, "Salient regions for query by image content," in *Image and Video Retrieval: Third International Conference, CIVR 2004*, P. Enser, Y. Kompatsiaris, N. E. O'Conner, A. F. Smeaton, and A. W. M. Smeulders, eds., pp. 317–325, Springer, (Dublin, Ireland), July 2004.
13. D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision* **60**, pp. 91–110, January 2004.
14. K. Mikolajczyk, *Detection of local features invariant to affine transformations.* PhD thesis, Institut National Polytechnique de Grenoble, France, 2002.
15. K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," in *International Conference on Computer Vision & Pattern Recognition*, **2**, pp. 257–263, June 2003.
16. M. Westmacott and P. Lewis, "An inverted index for image retrieval using colour pair feature terms," in *Proceedings of the SPIE Image and Video Communications and Processing Conference*, pp. 881–889, January 2003.
17. M. F. Porter, "An algorithm for suffix stripping," *Program* **14**(3), pp. 130–137, 1980.
18. L. Page and S. Brin, "The anatomy of a search engine," in *The 7th International WWW Conference (WWW 98)*, (Brisbane, Australia), April 1998.
19. E. Vincent and R. Laganière, "Detecting planar homographies in an image pair," in *Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis*, pp. 182–187, (Pula, Croatia), June 2001.
20. G. Pass, R. Zabih, and J. Miller, "Comparing images using color coherence vectors," in *Proceedings of ACM Multimedia*, pp. 65–73, 1996.
21. M. F. A. Fauzi and P. H. Lewis, "Query by fax for content-based image retrieval," in *Proceedings of the International Conference on Image and Video Retrieval*, pp. 91–99, Springer-Verlag, 2002.