Fast Content-Based Retrieval from Online Photo Sharing Sites

Gerald Schaefer and David Edmundson

Department of Computer Science, Loughborough University, Loughborough, U.K.
gerald.schaefer@ieee.org, d.edmundson@lboro.ac.uk

Abstract. Literally billions of images have been uploaded to photo sharing sites since their inception, comprising a staggering wealth of visual information. However, effective tools for querying these collections are rare and keyword based. Since users rarely annotate their images, this approach is only of limited use. Content-based image retrieval (CBIR) extracts features directly from images and bases searches on these features. However, conventional CBIR approaches require a dedicated system that performs feature extraction during photo upload and a database system to store the features, and are hence not available to the average user. In this paper, we present a very fast content-based retrieval method that performs feature extraction on-the-fly during the retrieval process and thus can be employed client-side on images downloaded from photo sharing sites such as Flickr.

Our approach is based on the fact that images uploaded to Flickr are stored in a JPEG format optimised to minimise disk space and bandwidth usage. In particular, we exploit the optimised Huffman compression tables, which are stored in the JPEG headers, as image descriptors. Since, in contrast to other approaches, we thus have to read only a fraction of the image file and similarity calculation is of low complexity, our approach is extremely fast as demonstrated by the bandwidth used to retrieve images from the Flickr photo sharing site. We also show that nevertheless retrieval performance is comparable to CBIR using colour histograms which is at the core of many CBIR systems.

1 Introduction

Visual information is becoming more and more important and at a rapid rate. One of the prime examples where this can be observed is the exponential growth of images uploaded to photo sharing sites such as Flickr[1] where literally billions[2] of images are available. Although this clearly comprises a staggering amount of visual information, retrieval tools are typically rather simplistic and implemented as simple keyword based searches. Since users rarely annotate images [1], searches thus often do not return useful images.

[1] http://www.flickr.com

[2] http://blog.flickr.net/en/2011/08/04/6000000000/

R. Huang et al. (Eds.): AMT 2012, LNCS 7669, pp. 278–287, 2012.

Content-based image retrieval (CBIR) techniques [2,3,4,5,6] extract image features such as colour or texture features directly from the image data and can hence be employed also when no textual annotations are available. However, conventional CBIR requires the features to be extracted during image upload and thus a dedicated system as well as a database to store the feature data. As such, this approach would not be available to users who want to perform content-based queries from a site such as Flickr.

While a client-side solution is in principle possible, this would require all image data to be downloaded during the retrieval process, then the features to be calculated and finally retrieval based on these features to be performed. Unfortunately, in particular the first step is prohibitive in terms of the time it requires, even with today's bandwidth provisions.

In this paper, we therefore present a very fast method to perform content-based retrieval of images stored on the Flickr photo sharing site. Since our method relies on data contained in the header of JPEG images, only a fraction of the whole image file needs to be downloaded during the retrieval process, leading to a significant speedup in terms of retrieval time. Experimental results confirm that while this enables interactive online retrieval for a database uploaded to Flickr, retrieval accuracy is comparable to classical image retrieval using colour histograms and colour based retrieval in the JPEG compressed domain.

The remainder of the paper is organised as follows. Section 2 briefly introduces CBIR concepts, while Section 3 describes the JPEG image compression algorithm. Section 4 then describes our contribution which allows for very fast JPEG image retrieval from online photo sharing sites. Experimental results are given in Section 5, and Section 6 concludes the paper.

2 Content-Based Image Retrieval

Colour was the first type of feature exploited for CBIR. Swain and Ballard [7] introduced the use of colour histograms, which record the frequencies of colours in the image, to describe images in order to perform object recognition and image retrieval. As similarity measure they introduced histogram intersection which quantifies the overlap between two histograms and can be shown to be equivalent to an L_1 norm. Starting from early CBIR systems such as QBIC [8] or Virage [9], colour histograms or related features have been at the core of many CBIR implementations [2]. Other kinds of features that are commonly used for CBIR include spatial colour, texture and shape features for which various types of algorithms have been suggested in the literature [2,3].

Almost all CBIR techniques operate in the pixel domain; that is they are calculated from pixel values in the images. On the other hand, virtually all images are stored in compressed form, most commonly in JPEG format, to reduce storage and bandwidth requirements. Consequently, for feature calculation the images need to be fully decompressed to first arrive at pixel data leading to a computational overhead during feature generation.

Faster feature extraction is possible using compressed domain CBIR algorithms [10,11] which operate directly on compressed image data and hence require only partial decoding. However, these methods are still not sufficiently fast for online image retrieval, e.g. retrieval from highly dynamic databases from the web such as photo sharing sites, or without access to internal feature databases. Here, pre-calculated features are not available (to the query process), and therefore must be calculated client-side "on-the-fly" during retrieval. As this must be conducted for every image in the dataset, the time taken is significant as the complete image file needs to be read in and partial decompression conducted.

3 JPEG Image Compression

JPEG [12] is not only the most popular image compression technique[3], it has also been adopted as an ISO standard for still picture coding. JPEG compression is based on the discrete cosine transform (DCT), a derivative of the discrete Fourier transform. First, an (RGB) image is usually converted into the YCbCr space. The reason for this is that the human visual system is less sensitive to changes in the chrominance (Cb and Cr) channels than in the luminance (Y) channel. Consequently, the chrominance channels can be downsampled by a factor of 2 without significantly reducing image quality, resulting in a full resolution Y and downsampled Cb and Cr components.

The image is then divided (each colour channel separately) into 8 × 8 pixel sub-blocks and DCT is applied to each such block. The 2-d DCT for an 8 × 8 block $f_{xy}, x, y = 0 \ldots 7$ is defined as

$$F_{uv} = \frac{C_u C_v}{4} \sum_{x=0}^{7} \sum_{y=0}^{7} f_{xy} \cos\left(\frac{(2x+1)u\pi}{16}\right) \cos\left(\frac{(2y+1)v\pi}{16}\right) \tag{1}$$

with $C_u, C_v = 1/\sqrt{2}$ for $u, v = 0$, $C_u, C_v = 1$ otherwise.

DCT has energy compactification close to optimal for most images which means that most of the information is stored in a few, low-frequency, coefficients (due to the nature of images which tend to change slowly over image regions). Of the 64 coefficients, the one with zero frequency (i.e., F_{00}) is termed "DC coefficient" and the other 63 "AC coefficients". The DC term describes the mean of the image block, while the AC coefficients account for the higher frequencies. As the lower frequencies are more important for the image content, higher frequencies can be neglected which is performed through a (lossy) quantisation step that crudely quantises higher frequencies while preserving lower frequencies more accurately.

The AC and DC components of the image are stored in separate streams for each colour channel. While the AC coefficients are zig-zag, run-length and entropy encoded, the DC stream is differentially encoded. That is, rather than storing the actual DC values, the differences between DC values are saved. As

[3] Up to 95% of all images on the web are JPEG images [13].

DC values range in $[-1024; 1024]$, the range of possible differences between DC components is $[-2048; 2048]$. The difference values are stored as two components: the first component, known as the "DC code", represents the size of the change in number of bits, while the second component stores the actual difference between the DC blocks.

The DC codes are then entropy encoded, and Huffman coding [14] is employed for this step. A standard Huffman table is provided by the JPEG group and is commonly used to compress images.

4 Fast JPEG Retrieval from Online Photo Sharing Sites

While normally JPEG images need to be fully decompressed to arrive at pixel data and hence enable feature calculation, it is also possible to derive image features directly in the JPEG compressed domain [15,16]. These features are based on the DCT coefficients expressed in Equation (1) and hence still require partial decoding of the file to undo the entropy, run-length and differential coding stages as well as the reversal of the quantisation step. To further reduce the computational load, it is possible to utilise only the DC stream (and hence ignore AC data), e.g. to calculate a colour histogram of the image [17,13]. For online image retrieval this approach is however still not nearly fast enough as the overhead associated with downloading the complete image files is dominant. Consequently, it appears to be impossible to perform content-based image retrieval from photo sharing sites such as Flickr, by applying CBIR methods on a set of images retrieved from there, within a reasonable time frame.

In this paper, we present a method that allows for exactly this kind of image retrieval. Our approach performs extremely fast image retrieval based solely on information contained in the header of JPEG files. Our method also utilises only the DC data of JPEG images which, since the DC term represents the average of an 8×8 block, corresponds to a downsampled version of the image [17]. That retrieval based on subsampled images can give similar results to retrieval based on full resolution pictures has already been demonstrated in [7], and we hence do not necessarily eliminate crucial information when using only the DC stream.

Now looking closer at how DC data is encoded (see Section 3), we see that rather than the DC terms themselves the differences between neighbouring DC terms (that is differences between the averages of neighbouring blocks) are stored. This data is however directly useful for content-based retrieval as has been shown in [18]. This comes from the fact that differences of neighbouring DC coefficients essentially give a description of the image gradient and hence a feature that encapsulates image variance as well as edges and uniform areas (the latter giving 0-differences between neighbouring blocks). Interestingly also, a histogram of these simple differences has been shown [18] to provide better CBIR performance than other, more sophisticated features such as the LBP operator [19] applied to DC terms as in [17].

Speedwise this approach is fast but not significantly so compared to other JPEG compressed domain algorithms [15,16] as it still requires entropy decoding and hence also reading in of the complete image files. We therefore go one step further and present an algorithm that is based solely on data available in the header of JPEG images.

Following the JPEG compression scheme, the DC differences are then entropy coded using Huffman coding [14]. While, as mentioned, standard tables (one for luminance and one for chrominance channels) defined by the JPEG group are typically used for this assignment, the Huffman tables can also be optimised to lead to increased compression in an image adaptive way. For this optimisation, a frequency table is built for each DC code and each code is then assigned a unique prefix code that assigns shorter bit strings to the most frequently occurring DC Codes (i.e. the most occurring DC differences), and longer strings to those less common. This optimisation is commonly employed by major image websites such as Flickr or Google Images[4]. For the former it is performed during photo upload and does not require any user intervention, while all JPEG images that can consequently be downloaded from Flickr contain image adapted Huffman tables.

The Huffman tables contain statistical information about DC difference occurrences in the image and hence provide an approximation of the DC difference histograms that were used in [18]. This indeed is the core of our idea and we therefore employ the optimised Huffman tables directly as image features. Since they are stored in the header of JPEG files, only a fraction of the image file needs to be downloaded for feature extraction. Our methods should hence be extremely fast for online image retrieval which we will seek to demonstrate in Section 5.

To compare the Huffman tables of two JPEG images, we use the length in bits of the prefix code assigned to each DC code directly as a feature. To calculate a distance representing dissimilarity between images, we utilise the L_1 norm between the feature vectors. Thus, for two images I_1 and I_2 with bitlength vectors f_{I_1} and f_{I_2}, the dissimilarity between the images is calculated as

$$d(I_1, I_2) = \sum_{i=1}^{12} |f_{I_1}(i) - f_{I_2}(i)|, \tag{2}$$

where i indicates the DC code of which there are at most 12 in a JPEG Huffman table. In cases where an entry does not exist in the Huffman table, the corresponding bitlength in the feature vector is set to the maximum over all other bitlengths plus 1 to indicate that the corresponding DC code appears even less frequently than all the other ones.

To incorporate both intensity and colour information, distances between luminance and chrominance DC tables are calculated and added to give a combined distance measure.

[4] `http://images.google.com`

5 Experimental Results

In our experiments we performed retrieval of images from the photo sharing site Flickr. As mentioned, all Flickr images contain optimised Huffman tables which we can hence readily exploit for our presented image retrieval algorithm. We uploaded the UCID dataset [20], a database of about 1400 images, to Flickr and wrote an application that, given a Flickr query image, performs online retrieval on the UCID images hosted on Flickr. That is, it performs feature extraction both on the query image and all database images, before calculating distances between query and model images and presenting the images to the user sorted by visual similarity. Our Flickr Browser application is available at `http://www-staff.lboro.ac.uk/~cogs/software/flickr-browser/`.

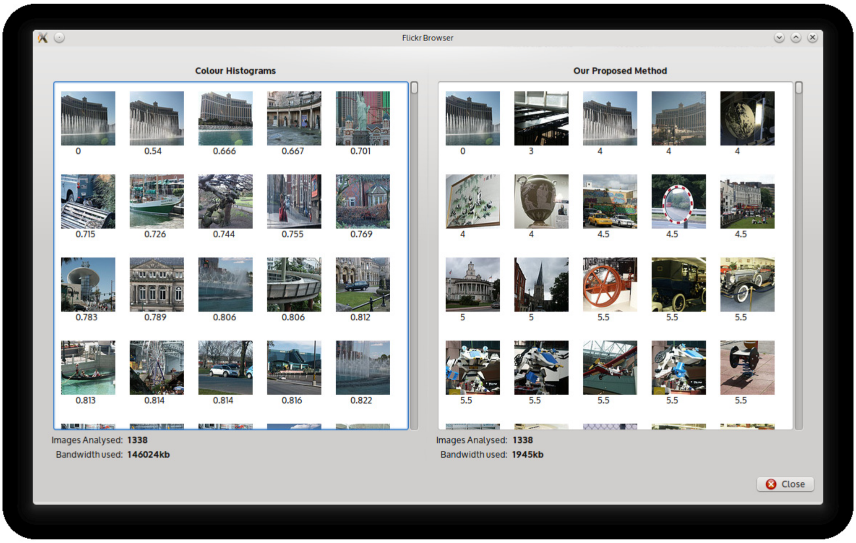
For comparison, we also integrated two other methods into our application, one from the pixel domain and one from the compressed domain. For the former we chose colour indexing [7] based on RGB colour histogram due to its popularity and wide spread in CBIR systems. As compressed domain method, we use Jiang *et al.*'s direct content access algorithm [13] where the authors exploited the fact that the average colour of an 8 × 8 block can be obtained directly from the DC component of the block. While they also showed that it is possible to extract the average value for 4 × 4 blocks through simple operators involving the first 3 AC components, they found this to be less effective, and we therefore also use DC based colour histograms calculated in YCbCr colour space (i.e. the colour space JPEG images are usually encoded in).

In Table 1 we give the average bandwidth required (as averages of more than 200 separate queries) for all implemented methods. We chose the bandwidth to give a measure that is independent of network load and connection speed; clearly low bandwidth indicates high retrieval speed while high bandwidth requirements equate to low speed retrieval processes. Since Flickr stores images at multiple resolutions, we give results for *large* images which are 500 pixels along the larger dimension and are hence of similar size to the original uploaded images as well as for *small* images which are thumbnails with a maximal dimension of 100 pixels.

Looking at Table 1, we can see that colour indexing based on *large* Flickr images is clearly prohibitive in terms of required bandwidth as it results in a download of almost 150 MB of image data. Using the *small* images significantly reduces the bandwidth requirement to under 6 MB, however it is also known

Table 1. Average bandwidth requirements of an online query for all CBIR methods

	bandwidth [KB]
colour indexing [7] (large)	146,012
colour indexing [7] (small)	5,751
Jiang *et al.* [13] (large)	146,012
Jiang *et al.* [13] (small)	5,751
Huffman table (large)	1,398



Images Analysed: 1338
Bandwidth used: 146024kb

Images Analysed: 1338
Bandwidth used: 1945kb

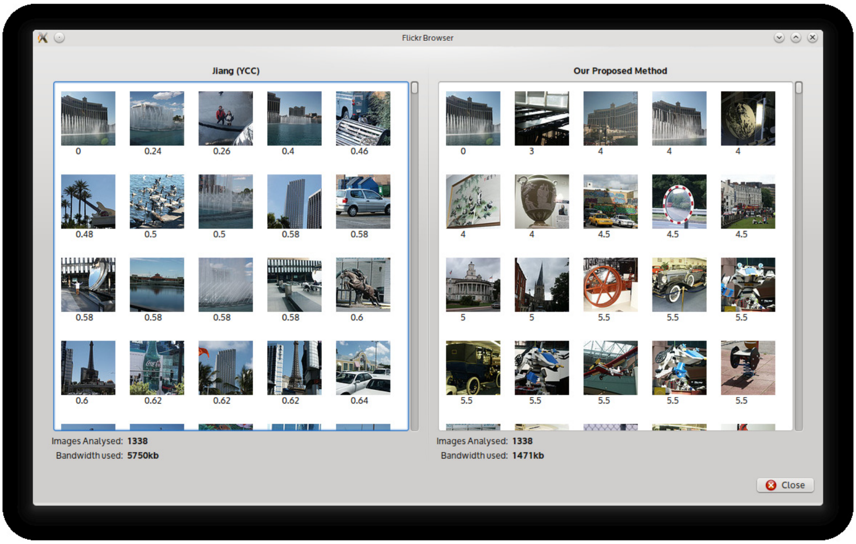
Fig. 1. Sample query on the UCID dataset based on colour histograms (left) and our Huffman method (right). The query image is the image on the top left (and hence also the first ranked retrieval) while the top 20 retrieved images for both methods are shown.

that retrieval based on thumbnails typically does not match that based of the original images.

While Jiang *et al.*'s method operates in the compressed domain of JPEG and calculates colour histograms based on DCT data, we can see that this does not affect the amount of data that needs to be downloaded since the complete image file needs to be read to arrive at the coefficient data. While the feature calculation itself is about 7 times faster than that for colour indexing [15] this speedup is negligible in comparison with the time required for downloading the images.

Finally, we look at the results of our presented method. Since we require only the JPEG headers, the bandwidth requirements are much lower even for the *large* images for which we report the results here. With less than 1.5 MB of data downloaded this corresponds to a reduction of more than 3 orders of magnitude compared to the other methods, and still leading to a more than 4-fold speedup when comparing it to retrieval of the *small* images for the other algorithms. Furthermore, as we utilise the Huffman tables directly as image features, feature calculation itself is not necessary which results in a further speedup compared to other approaches.

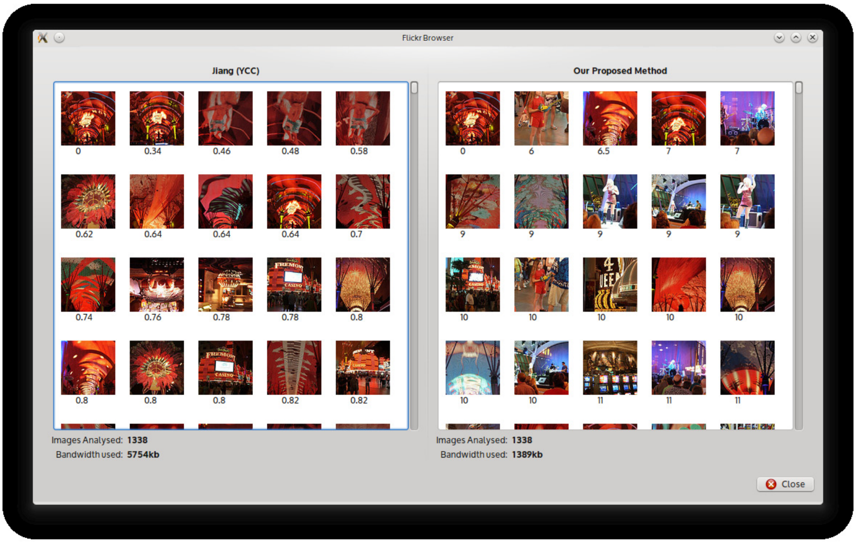
That we do not compromise in terms of retrieval accuracy is demonstrated in Figures 1 to 3. Figure 1 gives a retrieval example and comparison between colour indexing (based on *large* images) and our method. While it is clear that the bandwidth required for our algorithm is significantly lower (our application



Images Analysed: 1338
Bandwidth used: 5750kb

Images Analysed: 1338
Bandwidth used: 1471kb

Fig. 2. Retrieval results of the same query as in Figure 1 using Jiang *et al.*'s method and our approach



Images Analysed: 1338
Bandwidth used: 5754kb

Images Analysed: 1338
Bandwidth used: 1389kb

Fig. 3. Another retrieval example using Jiang *et al.*'s method and our Huffman table algorithm

actually performs retrieval for both methods concurrently which results in the grid for our method to be "filled" much faster than for other methods), we still get good retrieval results with both approaches retrieving two matches in the top 20 ranked images.

Figure 2 shows results for the same query using Jiang *et al.*'s method (on *small* images) and our approach. Clearly, for the latter the results are the same as in Figure 1[5], while for the compressed domain colour histogram algorithm none of the top 20 images contain the building of the query image.

Another retrieval example is given in Figure 3, again for Jiang *et al.*'s and our algorithm. Clearly, for the chosen query image a colour-based approach works very well while our approach, based on differences between neighbouring blocks lacks somewhat behind yet still retrieves some relevant images.

6 Conclusions

Using current algorithms, content-based image retrieval in an online fashion from photo sharing sites without access to internal feature databases and hence requiring client-side feature extraction and comparison does not allow for interactive retrieval times since the complete files of all images need to be downloaded during the query process. In this paper, we have presented a fast method for retrieving images from Flickr based on information contained solely in the header of JPEG files. In particular, we exploit image adaptive Huffman tables, which are generated during photo upload to Flickr, as image features and hence require only a fraction of the files to be downloaded during retrieval resulting, as demonstrated, in a significant reduction of bandwidth requirements while still providing good retrieval performance.

References

1. Rodden, K.: Evaluating Similarity-Based Visualisations as Interfaces for Image Browsing. PhD thesis, University of Cambridge Computer Laboratory (2001)
2. Smeulders, A., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-based image retrieval at the end of the early years. IEEE Trans. Pattern Analysis and Machine Intelligence 22, 1249–1380 (2000)
3. Datta, R., Joshi, D., Li, J., Wang, J.Z.: Image retrieval: Ideas, influences, and trends of the new age. ACM Computing Surveys 40, 1–60 (2008)
4. Schaefer, G.: Mining Image Databases by Content. In: Fernandes, A.A.A., Gray, A.J.G., Belhajjame, K. (eds.) BNCOD 2011. LNCS, vol. 7051, pp. 66–67. Springer, Heidelberg (2011)
5. Schaefer, G.: Content-Based Image Retrieval: Some Basics. In: Czachórski, T., Kozielski, S., Stańczyk, U. (eds.) Man-Machine Interactions 2. AISC, vol. 103, pp. 21–29. Springer, Heidelberg (2011)

[5] Some equi-distant images appear in a slightly different order. The actual bandwidth results vary to a certain degree due to different sizes of the query images as well as due to other concurrent applications and difficulties in measuring the "true" bandwidth.

6. Schaefer, G.: Content-Based Image Retrieval: Advanced Topics. In: Czachórski, T., Kozielski, S., Stańczyk, U. (eds.) Man-Machine Interactions 2. AISC, vol. 103, pp. 31–37. Springer, Heidelberg (2011)
7. Swain, M., Ballard, D.: Color indexing. Int. Journal of Computer Vision 7, 11–32 (1991)
8. Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., Yanker, P.: Query by image and video content: The QBIC system. IEEE Computer 28, 23–32 (1995)
9. Bach, J., Fuller, C., Gupta, A., Hampapur, A., Horowitz, B., Humphrey, R., Jain, R.: The Virage image search engine: An open framework for image management. In: Storage and Retrieval for Image and Video Databases. Proceedings of SPIE, vol. 2670, pp. 76–87 (1996)
10. Mandal, M., Idris, F., Panchanathan, S.: A critical evaluation of image and video indexing techniques in the compressed domain. Image and Vision Computing 17, 513–529 (1999)
11. Schaefer, G.: Content-based retrieval of compressed images. In: International Workshop on Databases, Texts, Specifications and Objects, pp. 175–185 (2010)
12. Wallace, G.: The JPEG still picture compression standard. Communications of the ACM 34, 30–44 (1991)
13. Jiang, J., Armstrong, A., Feng, G.: Direct content access and extraction from JPEG compressed images. Pattern Recognition 35, 1511–2519 (2002)
14. Huffman, D.: A method for the construction of minimum redundancy codes. Proceedings of the Institute of Radio Engineers 40, 1098–1101 (1952)
15. Edmundson, D., Schaefer, G.: Performance comparison of JPEG compressed domain image retrieval techniques. In: IEEE Int. Conference on Signal Processing, Communications and Computing (2012)
16. Edmundson, D., Schaefer, G.: An overview and evaluation of JPEG compressed domain retrieval techniques. In: 54th International Symposium ELMAR (2012)
17. Schaefer, G.: JPEG image retrieval by simple operators. In: 2nd International Workshop on Content-Based Multimedia Indexing, pp. 207–214 (2001)
18. Schaefer, G., Edmundson, D.: DC stream based JPEG compressed domain image retrieval. In: 8th Int. Conference on Active Media Technology (2012)
19. Ojala, T., Pietikäinen, M., Harwood, D.: A comparative study for texture measures with classification based on feature distributions. Pattern Recognition 29, 51–59 (1996)
20. Schaefer, G., Stich, M.: UCID - An Uncompressed Colour Image Database. In: Storage and Retrieval Methods and Applications for Multimedia. Proceedings of SPIE, vol. 5307, pp. 472–480 (2004)