

Efficient and effective JPEG compressed domain image retrieval

Gerald Schaefer

Abstract Content-based image retrieval (CBIR) has been an active research area for more than two decades. However, while almost all images are stored in compressed form, the vast majority of CBIR algorithms operate in the (uncompressed) pixel domain. This not only leads to a computational overhead for feature calculation, it can also be demonstrated that image compression has a negative effect on retrieval accuracy, especially at extreme compression rates. In this paper, we present efficient and effective CBIR techniques that operate directly in the compressed domain, hence not requiring full decompression for feature extraction. In particular, we focus on the JPEG domain since most images are compressed using JPEG. We show how CBIR features can be extracted from DCT coefficients, from differentially coded DC data, and from tuned information contained in the JPEG headers.

Key words: content-based image retrieval, image databases, compressed domain retrieval, JPEG, discrete cosine transform.

1 Introduction

Visual information, in particular in form of images, is increasing at a rapid rate. However, retrieving images faces several challenges, in particular since users rarely annotate images [10]. Content-based image retrieval (CBIR) [16, 1, 14, 13] allows to search for images based on features, describing e.g. colour or texture properties, extracted directly from the image data.

Almost all CBIR algorithms operate in the pixel domain and therefore require decompressing the image prior to feature extraction. This not only leads to a computational overhead for feature calculation, it can also be demonstrated that image compression has a negative effect on retrieval accuracy, especially at extreme compression rates [12]. Significantly faster feature calculation is possible by extracting image features directly in the compressed domain of images [7]. JPEG is the most popular image compression algorithm and it is reported that up to 95% of the images

G. Schaefer

Department of Computer Science, Loughborough University, U.K., e-mail: gerald.schaefer@ieee.org

on the web are JPEG images [8]. JPEG is a lossy image compression technique that splits the image into blocks of 8×8 pixels and applies the discrete cosine transform (DCT) to separate high and low frequency data, which facilitates discarding visually less important high frequency information.

In this paper, we present efficient and effective CBIR techniques that operate directly in the compressed domain of JPEG. In particular, we show how CBIR features can be extracted from DCT coefficient data, from differentially coded DC data, and from tuned information contained in the JPEG headers.

2 JPEG image compression

JPEG [17] is currently the most popular image compression technique and has been adopted as an ISO standard for still picture coding. It 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 colour 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 the DCT applied to each such block. The 2-d DCT for an 8×8 block $f_{xy}, x, y = 0 \dots 7$ is defined as

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

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 DC and AC components of the image are stored in separate streams. The DC stream is differentially encoded, while the AC stream is run-length coded. Finally, Huffman coding [6] is applied to both DC and AC data to maximise compression.

3 JPEG retrieval from DCT data [11]

Retrieval of JPEG images is often performed based on DCT coefficients. Using DCT coefficients directly avoids having to apply the computationally costly inverse discrete cosine transform to arrive back at pixel information.

We can also utilise that JPEG images are normally encoded in the YCbCr colour space rather than the RGB space itself. As in this colour space the achromatic and chromatic information are separated, we can derive texture information directly from the luminance (i.e. grayscale) Y channel, while calculating colour features from the Cb and Cr chromaticity channels.

After DCT, the DC terms hold the mean values of each image block. In other words, the DC components describe a lower resolution version of the original image. We can therefore directly calculate (colour) chromaticity (CbCr) histograms from the DC coefficients to describe the colour content of JPEG images. The colour content of two JPEG images can then be compared using the L_1 norm.

For texture information, similar to the colour index, we can use the DC terms only. In particular, we can apply the local binary pattern (LBP) operator [9] on the DC terms of the luminance channel. A 256-bin LBP histogram is thus generated which can also be compared using the L_1 norm.

Integrating colour and texture descriptors can be done by adding and weighting the relative scores acquired from the two methods. The weighting factors can be adjusted so as to give best results for a certain kind of images or application. Image retrieval is performed by calculating the matching scores for each image in the database and ranking the images by calculated similarity.

Clearly, this method is not the only possibility of performing image retrieval based on DCT coefficient data; a summary of other JPEG CBIR algorithms can be found in [4].

4 JPEG retrieval from differentially coded DC data [15]

As mentioned above, the DC stream is differentially coded, that is as differences between the DC values of (neighbouring) image blocks.

That differentially encoded DC data is directly useful for image retrieval. Since DC values characterise the average of image blocks, the differences between them describe important information about the image gradient, image texture and about edges in the image as well as about uniform image areas (where the differences would be 0 or close to it). We can therefore encode the DC difference data in form of a histogram over the whole image to obtain a feature useful for CBIR.

Since there are three colour channels (Y, Cb, and Cr), we can build a histogram for each of the channels. However, these histograms need to be carefully defined to be useful for image retrieval. As the DC values are in the range $[-1024; 1024]$, the possible differences between the DC components are in $[-2048; 2048]$. However, lower differences are statistically much more likely than higher ones. Therefore,

the boundaries of the histogram bins are set to $\pm(2^m)$, $m = \{0, 1, 2, \dots, 11\}$ with an additional bin for 0. That is, the first bin of the histogram encodes 0 differences, the second bin absolute differences of 1, the third absolute differences in $[2; 3]$, the fourth in $[4; 7]$, etc., and the last bin absolute differences in $[1024; 2048]$.

The histograms can then be normalised (to sum to 1) and compared using an L_1 norm. To integrate the three histograms, the calculated distances are summed up into a single measure, while weights of 0.6:0.2:0.2 for Y:Cb:Cr allow to put more emphasis on the luminance channel which is dominantly affected by image texture.

5 JPEG retrieval from header data [3, 5]

While retrieval based on DCT coefficient avoids the computationally expensive IDCT, the other compression steps, i.e. quantisation, differential/runlength coding and Huffman coding still need to be reversed. Using DC difference data directly accesses the compressed stream at an earlier stage but still needs to reverse the Huffman coding stage.

The ideal would be a technique that does not require any decompression at all. In fact, such a method can be found and is based on information that is readily available in the header of optimised JPEG images.

As mentioned, the DC and AC codes are entropy encoded using Huffman coding [6]. Standard Huffman tables are provided by the JPEG group and are commonly used to compress images. However, in order to achieve optimal compression the Huffman tables can be adapted to the image being compressed, which is commonly utilised by image repositories including Flickr¹ (during image upload) and Google Images² (for thumbnail images).

As the table optimisation process assigns the smallest codes to the most commonly occurring DC or AC codes, we can use the thus adapted Huffman tables as an indication of the frequencies of each code. Similar images should thus lead to similar Huffman tables and the Huffman tables can be used directly as CBIR features.

In order to compare the tables from two images, and hence to arrive at a measure of (dis)similarity between the images, we can use the bitlengths of the Huffman codes and employ the L_1 norm to compare the feature vectors. Information from both the DC and the AC tables can be exploited in this way. Since the tables are stored in the header of JPEG files, not only is no explicit feature extraction necessary but also only a small part of the image file needs to be retrieved which makes it particularly useful for online image retrieval [2].

¹ <http://www.flickr.com>

² <http://images.google.com>

6 Conclusions

In this paper, we have shown how features useful for content-based image retrieval can be extracted from JPEG encoded images directly in the compressed domain. In particular, we have demonstrated how features based on DCT coefficients can be employed, how the differentially coded DC stream can be used for retrieval, and how optimised Huffman tables residing in the header of JPEG files can be utilised directly as CBIR features. For further details on the presented methods, we refer the reader to the given references.

References

1. R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40(2):1–60, 2008.
2. D. Edmundson and G. Schaefer. Efficient and effective online image retrieval. In *IEEE Int. Conference on Systems, Man, and Cybernetics*, 2012.
3. D. Edmundson and G. Schaefer. Fast JPEG image retrieval using optimised Huffman tables. In *21st Int. Conference on Pattern Recognition*, pages 3188–3191, 2012.
4. D. Edmundson and G. Schaefer. Performance comparison of JPEG compressed domain image retrieval techniques. In *IEEE Int. Conference on Signal Processing, Communications and Computing*, 2012.
5. D. Edmundson and G. Schaefer. Very fast image retrieval based on JPEG Huffman tables. In *2nd IAPR Asian Conference on Pattern Recognition*, 2013.
6. D. A. Huffman. A method for the construction of minimum redundancy codes. *Proceedings of the Institute of Radio Engineers*, 40:1098–1101, 1952.
7. F. Idris and S. Panchanathan. Storage and retrieval of compressed images. *IEEE Trans. Consumer Electronics*, 41(3):937–941, 1995.
8. J. Jiang, A. Armstrong, and G. C. Feng. Direct content access and extraction from JPEG compressed images. *Pattern Recognition*, 35(11):1511–1519, 2002.
9. T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study for texture measures with classification based on feature distributions. *Pattern Recognition*, 29:51–59, 1996.
10. K. Rodden. *Evaluating Similarity-Based Visualisations as Interfaces for Image Browsing*. PhD thesis, University of Cambridge Computer Laboratory, 2001.
11. G. Schaefer. JPEG image retrieval by simple operators. In *2nd International Workshop on Content-Based Multimedia Indexing*, pages 207–214, 2001.
12. G. Schaefer. Does compression affect image retrieval performance? *Int. Journal of Imaging Systems and Technology*, 18(2–32):101–112, 2008.
13. G. Schaefer. Content-based image retrieval: Advanced topics. In *2nd Int. Conference on Man-Machine Interactions*, volume 103 of *Advances in Intelligent and Soft Computing*, pages 31–37, 2011.
14. G. Schaefer. Content-based image retrieval: Some basics. In *2nd Int. Conference on Man-Machine Interactions*, volume 103 of *Advances in Intelligent and Soft Computing*, pages 21–29, 2011.
15. G. Schaefer and D. Edmundson. DC stream based JPEG compressed domain image retrieval. In *8th Int. Conference on Active Media Technology*, 2012.
16. 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. Pattern Analysis and Machine Intelligence*, 22(12):1249–1380, 2000.
17. G. K. Wallace. The JPEG still picture compression standard. *Communications of the ACM*, 34(4):30–44, 1991.