RECOMPRESSING IMAGES TO IMPROVE IMAGE RETRIEVAL PERFORMANCE

David Edmundson and Gerald Schaefer

Department of Computer Science, Loughborough University, Loughborough, U.K.

ABSTRACT

Virtually all images are stored in compressed form, most in (lossy) JPEG format. Compressing images however has been shown to cause a small but not negligible drop in performance for content-based image retrieval (CBIR) algorithms. In this paper, we show that it is possible to reverse this performance drop. We achieve this by what might at a first glance seem counter-intuitive, namely by compressing the images even more. In detail, what we perform is recompressing images (or rather re-quantising the DCT coefficients) to their lowest common image quality setting. We demonstrate, on a benchmark image retrieval database and using standard CBIR algorithms, that this results in improved image retrieval performance rivalling that of running the algorithms on uncompressed data.

***Index Terms*—** Content-based image retrieval, image compression, retrieval performance, JPEG.

1. INTRODUCTION

With the ever growing number of images, effective and efficient techniques for managing image collections are highly sought after. To that extent, over the past two decades content-based image retrieval (CBIR) has been an active research area leading to significant progress in this field [1, 2].

In practice, virtually all images are stored in compressed form in order to save bandwidth and storage resources. Of all image compression techniques, JPEG [3] is clearly the dominant one with up to 95% of images on the web being compressed in JPEG format [4].

JPEG is a lossy format, which means that during compression some of the visually less important information is discarded in order to obtain better compression rates. This is achieved through the application of a quantisation step on the derived DCT (discrete cosine transform) data. The balance between image quality and compression ratio is governed by a (user-defined) "q-factor", a scalar (in $[0; 100]$) that allows finer and coarser quantisation respectively.

In the context of CBIR, it has been shown that this loss of information leads to a small but not negligible drop in retrieval performance of common CBIR algorithms, especially for lower q-factors (i.e., lower image quality) [5, 6].

In this paper, we propose a method to counter this performance drop. We notice that when images are compressed to the same quality setting, retrieval performance is close to that achieved on uncompressed images. Building upon this, our method essentially recompresses images to the lowest common q-factor before feature calculation. Since by doing so, we discard even more of the original image information, this might at first seem to do more harm than good. However, since we thus explicitly transform image to a common image quality (and hence, common quantisation effects), we actually improve retrieval performance as extensive experiments on a benchmark dataset [7] and common CBIR algorithms [8, 9] demonstrate.

2. THE JPEG COMPRESSION SCHEME

JPEG [3] is currently the most popular image compression technique and has also 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 space. The reason for this is that the human visual system is less sensitive to changes in the chrominance (Cb and Cr) than in the luminance (Y) channel. Consequently, the chrominance channels can be downsampled by a factor of 2 without sacrificing too much image quality thus resulting in a full resolution Y and downsampled Cb an Cr components.

The image is then divided (each colour channel separately) into 8×8 pixel sub-blocks and DCT applied to each such block. The 2-d DCT for an 8×8 block $f_{xy}, x, y = 1 \dots 7$ is defined as

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

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.

The way JPEG is doing this is by applying a quantisation step that crudely quantises higher frequencies while preserving lower frequencies more accurately. In particular, two 8×8 quantisation tables are used, one for luminance, one for chrominance channels, and the DCT coefficients are divided by the factors in the tables, resulting in

$$\hat{F}_{uv} = round(F_{uv}/Q_{uv}) \tag{2}$$

where $\hat{F}_{uv}$ is the quantised DCT coefficient, and Q_{uv} the corresponding entry in the quantisation table. Clearly, this is a lossy procedure and thus not uniquely reversible.

The compression ratio (and consequently the image quality) is governed by a "q-factor" in the range $[0; 100]$ which essentially scales the quantisation table entries according to

$$\hat{Q}_{uv} = \begin{cases} round(Q_{uv} * (5000/q)/100) & \text{if } q < 50 \\ round(Q_{uv} * (200 - q * 2)/100) & \text{if } q \geq 50 \end{cases} \tag{3}$$

Thus, the lower the q-factor, the more information is discarded and hence the lower the image quality.

After quantisation, the DC terms are differentially coded since they change slowly over the image. The AC coefficients are ordered in a zig-zag fashion for each block and are run-length coded. Finally, both parts are entropy (Huffman) coded to maximise the compression efficiency.

3. CBIR UNDER COMPRESSION

Clearly, a (lossy) compressed image differs from its original, and as such also the features employed in CBIR will somewhat differ. This has been studied in detail in [5] where it was concluded that image compression does indeed have a small negative effect on retrieval performance, especially at lower image quality settings.

We reconfirm this by running retrieval experiments under different quality settings on the UCID database [7] which was also used in [5]. The dataset consists of 1338 uncompressed images of which 262 are query images for which matching ground truth images are defined. The ground truth thus allows to assess the performance of CBIR algorithms [10], and we chose the suggested modified average match percentile (AMP), defined as the average of $\text{MP}_Q = \frac{100}{S_Q} \sum_{i=1}^{S_Q} \frac{N - R_i}{N - i}$ over the whole dataset as performance measure (N is the number of images in the dataset, S_Q the number of model images for the given query, and R_i is the rank in which a correct model image was retrieved).

The dataset was compressed, using standard 4:2:0 subsampling, at 6 different q-factors: 100 (which closely resembles uncompressed data), 80, 50, 20, 10 and 5. We then run retrieval experiments at various compression levels by using the query images compressed with a certain q-factor together with model images compressed at another certain q-factor.

For our experimentation, we employ three commonly used (colour) CBIR techniques: colour indexing [8], and two of the MPEG-7 visual descriptors [9, 11], namely Scalable Colour (which builds a colour histogram in HSV space that is then encoded using a Haar wavelet transform of which the top n ($n = 256$ in our experiments) coefficents are used), and Colour Structure (which uses a colour histogram that is affected by local colour distribution by calculating a 256 bin histogram in HMMD colour space by moving an 8×8 sliding window over the image).

The results are given in Tables 1, 2, and 3. Each cell of the tables lists the achieved AMP based on the selected quality settings for query and model images, and also states the difference to the case where both model and query images are encoded with $q = 100$ (i.e., essentially uncompressed images).

M \ Q	100	80	50	20	10	5
100	90.79	90.74 -0.05	90.68 -0.11	90.69 -0.10	90.47 -0.32	87.33 -3.46
80	90.82 0.03	90.78 -0.01	90.75 -0.04	90.76 -0.03	90.54 -0.25	87.38 -3.41
50	90.77 -0.02	90.75 -0.04	90.73 -0.06	90.76 -0.03	90.57 -0.22	87.40 -3.39
20	90.79 0.00	90.77 -0.02	90.73 -0.06	90.78 -0.01	90.64 -0.15	87.46 -3.33
10	90.44 -0.35	90.46 -0.33	90.46 -0.33	90.56 -0.23	90.79 0.00	87.04 -3.75
5	88.12 -2.67	88.16 -2.63	88.19 -2.60	88.31 -2.48	88.74 -2.05	89.36 -1.43

Table 1. Retrieval performance results, in AMP, for colour indexing [8] under compression. Query images were compressed with q-factor Q, while model images were compressed using M.

M \ Q	100	80	50	20	10	5
100	92.69	92.75 0.06	92.49 -0.20	92.47 -0.22	89.22 -3.47	80.18 -12.51
80	92.37 -0.32	92.73 0.04	92.61 -0.08	92.61 -0.08	88.86 -3.83	79.14 -13.55
50	92.01 -0.68	92.61 -0.08	92.60 -0.09	92.66 -0.03	88.89 -3.80	78.99 -13.70
20	91.93 -0.76	92.62 -0.07	92.65 -0.04	92.84 0.15	89.73 -2.96	79.35 -13.34
10	89.93 -2.76	90.17 -2.52	90.06 -2.63	90.79 -1.90	92.64 -0.05	86.44 -6.25
5	87.00 -5.69	87.19 -5.50	87.54 -5.15	88.52 -4.17	91.30 -1.39	92.46 -0.23

Table 2. Retrieval performance results for MPEG-7 Scalable Colour [9] under compression.

M \ Q	100	80	50	20	10	5
100	94.41	94.26 -0.15	94.02 -0.39	93.59 -0.82	90.96 -3.45	79.89 -14.52
80	94.3 -0.11	94.37 -0.04	94.28 -0.13	93.89 -0.52	91.63 -2.78	80.38 -14.03
50	94.17 -0.24	94.36 -0.05	94.33 -0.08	94.0 -0.41	92.14 -2.27	80.99 -13.42
20	93.98 -0.43	94.22 -0.19	94.27 -0.14	94.09 -0.32	92.2 -2.21	81.07 -13.34
10	91.34 -3.07	91.81 -2.60	92.03 -2.38	91.19 -3.22	93.95 -0.46	87.10 -7.31
5	82.71 -11.70	83.02 -11.39	83.25 -11.16	81.24 -13.17	90.7 -3.71	93.01 -1.40

Table 3. Retrieval performance results for MPEG-7 Colour Structure [9] under compression.

We can see that all three tested algorithms are affected by image compression. Image retrieval drops, especially for lower quality settings. For example, image retrieval using colour histograms where query images are compressed using a q-factor of 5 and model images with a q-factor of 20 achieves an AMP of 88.31 compared to 90.79 for $q = 100$ for both query and model images.

A similar behaviour can be noticed for both the Scalable Colour and Colour Structure descriptors. However, for these retrieval methods performance drops even more significantly than for colour indexing. The average AMP difference between retrieval of virtually uncompressed images ($q = 100$) and compressed images is 0.97 for colour histograms, 3.08 for Scalable Colour and 4.05 for Colour Structure. Considering even the relatively small size of the UCID dataset this equates to having to browse through about 13 / 41 / 54 images more on average to arrive at the pictures of interest.

4. CBIR AFTER RECOMPRESSION

Looking at Tables 1 to 3 in more detail, we can however notice an interesting aspect of the results, namely that retrieval performance does not drop in all cases. In particular, looking along the diagonals of tables tables we see that retrieval performance is almost unaffected. What this means is that if both model and query images are compressed using the same q-factor, image retrieval is almost as good as for uncompressed images. This observation is the core of the idea we present in this paper. On the other hand, when the quality settings differ, performance drops, and the bigger the difference in terms of q-factors the bigger the drop. The reason for this is that the original DCT values get quantised using different(ly scaled) quantisation tables which causes the same original values to be mapped, after quantisation and reversing the quantisation, to different values hence leading to somewhat different image descriptors and thus lower retrieval performance.

Our proposed approach for improving retrieval performance for compressed images is simple yet effective. We want to ensure that when we encounter images with different image quality settings, we bring them to the same quality before extracting image features. Since clearly we cannot recover the original information that was discarded, the only way of doing so is to use the lower quality setting for all images. That is, we compress images even more - by recompressing those with higher settings to lower ones (lower quality images remain unchanged) - in order to improve retrieval performance.

When we say "recompress", we don't actually run through the complete compression procedure. Rather, all that we need to do is rescale the affected (quantised) DCT coefficients. We can identify the q-factor of in image by inspecting the quantisation tables of the JPEG header of the image. In fact, it is not required to extract the q-factor itself as the table entries themselves are sufficient for the rescaling operation.

For each entry where the quantisation tables differ, we identify the higher entry (corresponding to more quantisation and hence lower quality) and rescale the coefficient data of the other image (i.e., the higher quality image) according to

$$\hat{F}_{uv-\text{new}} = round\left(\frac{\hat{F}_{uv-\text{old}} * \hat{Q}_{uv-\text{higher}}}{\hat{Q}_{uv-\text{lower}}}\right) \qquad (4)$$

where $\hat{F}_{uv-\text{old}}$ are the original (quantised) coefficients of the higher quality image, $\hat{Q}_{uv-\text{higher}}$ are its quantisation table entries, and $\hat{Q}_{uv-\text{lower}}$ define the quantisation table of the lower quality image.

In practice, we perform this procedure on-line only for query images. For model images, in order to avoid a computational overhead during the retrieval process, we store different sets of features corresponding to different q-factor matches.

5. EXPERIMENTAL RESULTS

We run our algorithm on the same dataset and under the same settings as in Section 3, i.e. with differently paired q-factor setting on the UCID database and using colour histograms and the two MPEG-7 features. When the quality settings between query and model images differ, we "recompress" the images as detailed in Section 4.

The results are presented in Table 4 for colour histograms, Table 5 for MPEG-7 Scalable Colour, and Table 6 for MPEG-7 Colour Structure features. Each cell in the tables gives the achieved AMP under the selected compression settings together with the difference to the results with the same settings but without our proposed approach (i.e., the difference to the results in Tables 1 to 3).

Looking at the results, we can see that the proposed method does indeed provide a significant boost in terms of retrieval performance. In almost all cases, retrieval performance improves, in many instances up to or close to the best possible AMP, i.e. that achieved when $q = 100$ for both

M \ Q	100	80	50	20	10	5
100		90.78 0.04	90.73 0.05	90.78 0.09	90.79 0.32	89.35 2.02
80	90.78 0.04		90.71 -0.04	90.74 -0.02	90.75 0.21	89.36 1.98
50	90.73 -0.04	90.71 -0.04		90.76 0.00	90.79 0.22	89.09 1.69
20	90.79 0.00	90.82 0.04	90.73 0.00		90.79 0.15	89.19 1.73
10	90.79 0.35	90.72 0.26	90.79 0.33	90.82 0.26		87.04 0.00
5	89.35 1.23	89.17 1.01	89.38 1.19	88.40 0.09	88.74 0.00	

Table 4. Retrieval performance results, in AMP, for colour indexing [8] after recompression.

M \ Q	100	80	50	20	10	5
100		92.81 0.06	92.61 0.12	92.71 0.24	92.67 3.45	92.47 12.29
80	92.69 0.32		92.59 -0.02	92.3 -0.31	92.76 3.90	92.52 13.38
50	92.71 0.70	92.63 0.02		92.66 0.00	92.65 3.76	91.69 12.70
20	92.89 0.96	92.84 0.22	92.65 0.00		92.67 2.94	92.29 12.94
10	92.60 2.67	92.63 2.46	92.66 2.60	92.61 1.82		86.44 0.00
5	92.51 5.51	92.37 5.18	92.32 4.78	91.60 3.08	91.30 0.00	

Table 5. Retrieval performance results for MPEG-7 Scalable Colour [9] after recompression.

model and query images. We can also notice that in particular for the cases of strong compression (i.e., looking at the most left/right columns, respectively the top/bottom rows of the results tables), the change in terms of performance is remarkable. The only exception is when we are dealing with a combination of q-factors of 5 and 10; here the quantisation is extreme in both cases and recompression is not always able to accurately remap the calculated image features. On average, the AMP improves by 0.44 for colour histograms, by 3.19 for Scalable Colour, and by 3.34 for Colour Structure features. On the UCID dataset, this corresponds to the correct model images coming up 6 / 43 / 45 images "earlier" in the retrieval results.

6. CONCLUSIONS

In this paper, we have looked at the issue of content-based image retrieval under compression. We have confirmed that compression causes a drop in retrieval performance, especially at low quality settings. To address this issue, we have presented a simple yet effective approach to reverse the performance dip. We do this by recompressing (rescaling of JPEG DCT coefficients) of the images to a *lower* image quality setting, and demonstrate that this does indeed lead to improved retrieval performance. While the method presented is based on JPEG image compression, the main idea, i.e. that of increased compression for improved retrieval, is expected to be applicable to other coding techniques in a similar way.

M \ Q	100	80	50	20	10	5
100		94.36 0.10	94.35 0.33	94.06 0.47	93.94 3.98	93.01 13.12
80	94.37 0.07		94.27 0.01	93.86 -0.03	93.95 2.32	93.08 12.70
50	94.33 0.16	94.22 -0.14		94.0 0.00	93.95 1.81	92.86 1.87
20	94.12 0.14	93.97 -0.25	94.27 0.00		93.95 1.75	93.04 11.97
10	93.96 2.62	93.91 2.10	93.95 1.92	93.97 2.78		87.10 0.00
5	93.01 10.30	92.93 9.91	92.93 9.68	92.29 11.05	90.70 0.00	

Table 6. Retrieval performance results for MPEG-7 Colour Structure [9] after recompression.

References

[1] 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*, vol. 22, no. 12, pp. 1249–1380, 2000.

[2] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image retrieval: Ideas, influences, and trends of the new age," *ACM Computing Surveys*, vol. 40, no. 2, pp. 1–60, 2008.

[3] G.K. Wallace, "The JPEG still picture compression standard," *Commun. ACM*, vol. 34, pp. 30–44, 1991.

[4] J. Jiang, A.J. Armstrong, and G.-C. Feng, "Direct content access and extraction from JPEG compressed images.," *Pattern Recognition*, vol. 35, no. 11, pp. 2511–2519, 2002.

[5] G. Schaefer, "Does compression affect image retrieval performance?," *Int. Journal of Imaging Systems and Technology*, vol. 18, no. 2–32, pp. 101–112, 2008.

[6] E. Guldogan and M. Gabbouj, "DCT-based downscaling effects on color and texture-based image retrieval," in *IEE European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology*, 2004.

[7] G. Schaefer and M. Stich, "UCID - an uncompressed colour image database," in *Storage and Retrieval Methods and Applications for Multimedia*, 2004, pp. 472–480.

[8] M.J. Swain and D.H. Ballard, "Color indexing," *International Journal of Computer Vision*, vol. 7, pp. 11–32, 1991.

[9] T. Sikora, "The MPEG-7 visual standard for content description - an overview," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 11, no. 6, pp. 696–702, 2001.

[10] T. Deselaers, D. Keysers, and H. Ney, "Features for image retrieval: an experimental comparison," *Information Retrieval*, vol. 11, no. 2, pp. 77–107, 2008.

[11] M. Baştan, H. Çam, U. Güdükbay, and Ö. Ulusoy, "BilVideo-7: An MPEG-7-compatible video indexing and retrieval system," *IEEE MultiMedia*, vol. 17, no. 3, pp. 62–73, 2009.