Abstract—Content-based image retrieval (CBIR) extracts features from images to support image search. In this tutorial paper, we will review some of the basic CBIR algorithms that derive colour, texture and shape features, and will then show how image features can be extracted from the compressed domain, in particular from JPEG images, without the need of completely decompressing the images.

I. INTRODUCTION

Visual information, in particular in form of images, is becoming increasingly important and hence effective methods for dealing with these collections are highly sought after. Since most images are un-annotated [1], there has been a lot of interest in content-based image retrieval (CBIR) [2], [3], [4], [5] which extracts image features directly from the image data and uses these, coupled with a similarity measure, to query image collections. Image features typically describe the colour, texture, and shape “content” of the images, and in this paper we review several well-known descriptors that are employed in CBIR. In the second part of the paper, we highlight how CBIR features can be extracted directly from the compressed image domain without the need of fully decompressing the images.

II. CONTENT-BASED IMAGE RETRIEVAL

A. CBIR using colour features

Colour histograms [6] record the frequencies of colours in an image and represent the most widely used colour feature. To compare colour histograms and hence given an indication of visual similarity, one can perform histogram intersection to calculate

\[ d_{HIS}(I_1, I_2) = 1 - \sum_{k=1}^{N} \min(H_1(k), H_2(k)), \]

where \( H_1 \) and \( H_2 \) are the colour histograms of images \( I_1 \) and \( I_2 \), and \( N \) is the number of bins used for representing the histogram.

An alternative to the \( L_1 \) norm is to use the Euclidean distance (\( L_2 \) norm) between two histograms [7] which can also address the problem of possible false negatives due to slight colour shifts by taking into account the similarity between separate histogram bins. This can be expressed in a quadratic form distance measure as

\[ d_{QBC}(I_1, I_2) = (H_1 - H_2)A(H_1 - H_2)^T, \]

where \( H_1 \) and \( H_2 \) are again the two (vectorised) colour histograms, and \( A \) is an \( N \times N \) matrix containing inter-bin colour differences.

Rather than using colour histograms, a more compact descriptor for encoding the colour distribution of images is a colour signature. Colour signatures are a set \( \{(c_1, \omega_1), (c_2, \omega_2), \ldots, (c_m, \omega_m)\} \) where \( c_i \) define colour coordinates and \( \omega_i \) their associated weights (i.e. their relative frequencies in the image). A common way of deriving colour signatures for images is through a clustering process. Once colour signatures for images are determined, they are often compared by a metric known as the earth mover’s distance [8] which quantifies the amount of work required to transform one colour signature into the other one.

Simple colour features such as colour histograms are fast to compute, and are invariant to rotation and translation as well as robust to scaling and occlusions. On the other hand, they do not carry any information about the spatial distribution of the colours. Colour coherence vectors [9] were introduced as a method of introducing spatial information into the retrieval process. Colour coherence vectors consist of two histograms: one histogram of coherent and one of non-coherent pixels. Pixels are considered to be coherent if they are part of a continuous uniformly coloured area and the size of this area exceeds some threshold. Another approach to incorporate information on the spatial correlation between the colours present in an image are colour correlograms [10], which record joint probabilities of certain colours a certain distance away in the image.

B. CBIR using texture features

Co-occurrence matrices of an image \( I \) are defined by [11]

\[ C(i, j) = \sum_{x=1}^{n} \sum_{y=1}^{m} \begin{cases} 1 & \text{if } I(x, y) = i \text{ and } I(x + p, y + q) = j \\ 0 & \text{otherwise} \end{cases} \]

where \( i \) and \( j \) correspond to image (grey-level) values, and \( p \) and \( q \) are offset values. Typically several \( (p, q) \) pairs are
employed and from the corresponding co-occurrence matrices several statistical features are calculated to form a feature vector.

Local binary patterns (LBP) [12] are a simple yet effective texture analysis technique, which assigns, on a pixel basis, descriptors that describe the neighbourhood of that pixel in a binary fashion (by thresholding with the centre pixel value) and then forms a histogram of those descriptors. LBP can be calculated based on a circular neighbourhood, made rotation invariant through a simple mapping, while improved performance can be gained by focussing on so-called uniform patterns [13]. Based on the LBP idea, a large number of texture descriptors have been introduced in the literature [14].

C. CBIR using shape features

A simple yet effective shape feature can be derived by describing edge direction information [15]. Following an edge detection step using the Canny edge detector [16], a histogram of edge directions (typically in 5 degree steps) is generated, and then smoothed. Since it is a histogram feature, it can be compared using e.g. histogram intersection as in Eq. (1).

A similar approach is taken for the derivation of the MPEG-7 edge descriptor [17], while another possibility of extracting shape features from images is the extraction of image moments and moment invariants [18].

III. COMPRESSED-DOMAIN CBIR

CBIR features such as the ones discussed above are typically extracted from pixel data. On the other hand, images are almost always stored in compressed form to reduce storage and bandwidth requirements. Consequently, for feature extraction, images first need to be decompressed to arrive at pixel data. An alternative is to perform image retrieval directly in the compressed domain and hence allow faster feature extraction [19], [20].

While retrieval based on different compression algorithms (e.g. vector quantisation [21]) or even development of new compression schemes where the compressed data is visually meaningful [22] is possible, it has been reported that up to 95% of images on the web are JPEG images [23] and we hence focus on this compression technique here.

JPEG [24] employs the discrete cosine transform (DCT) on $8 \times 8$ image blocks $f_{xy}$, $x, y = 0 \ldots 7$

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

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. 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 (i.e. rather than storing the actual DC values, differences between DC values are saved), while the AC data is run-length coded. Finally, both streams are entropy coded using Huffman coding.

The main idea of most JPEG CBIR techniques is to derive image features directly from DCT coefficients, thus avoiding the computationally expensive inverse DCT [25]. In [26], only DC data is used to extract both colour and texture descriptors. A colour histogram is constructed from chromaticity (CbCr) DC data, while a texture descriptor is obtained by applying the LBP operator on the DC data of the luminance (Y) channel and generating a histogram of LBP descriptors. Both colour and texture histograms are compared using the $L_1$ norm, and the resulting scores weighted and added.

The approach in [27] extracts colour, texture and basic edge information from JPEG images. Colour information for $4 \times 4$ pixel blocks is approximated using the first 3 AC coefficients (i.e., $F_{01}, F_{10}$ and $F_{11}$) to build a colour histogram as in [23]. Two texture features are obtained. For the first one, an energy histogram is extracted for each block. Three features are calculated using different bands of energy coefficients. The second texture feature is based on the edge information in the block and is extracted using the sum of the coefficients which represent horizontal, vertical and diagonal energy shifts. The mean and standard deviation of these features across the images is then used resulting in a 12 key descriptor for image texture.

One can also make use of the fact that DC coefficients are differentially coded [28]. It is thus possible to calculate a histogram directly of these differences which captures information about the image gradient, image texture and edges as well as about uniform image areas (where the differences are 0 or close to it). This approach thus eliminates the need to perform differential decoding needed by other comparison methods.

Significantly faster retrieval is possible by employing tuned JPEG Huffman [29] or quantisation [30] tables. Since these are stored in the header of the image files, no decompression at all is necessary and only a small part of the file needs to be read.

IV. CONCLUSIONS

In this tutorial paper, we have given a brief introduction to several popular content-based image retrieval methods that extract colour, texture or shape information from images. Furthermore, we have discussed how similar features can be extracted from compressed images, in particular from images stored in JPEG format.

REFERENCES


