

CVPIC image retrieval based on block colour co-occurrence matrix and pattern histogram

Gerald Schaefer and Simon Lieutaud

School of Computing and Technology, The Nottingham Trent University
Nottingham, United Kingdom
gerald.schaefer@ntu.ac.uk

Guoping Qiu

School of Computer Science
University of Nottingham
Nottingham, United Kingdom

Abstract

Compressed domain image processing techniques are becoming increasingly important. Compressed domain retrieval allows the calculation of image features and hence content-based image retrieval (CBIR) to be performed directly on the compressed data without the need of decoding it beforehand. The Colour Visual Pattern Image Coding (CVPIC) technique represents a compression algorithm where the compressed form is directly meaningful. Based on CVPIC we introduce a compressed domain retrieval algorithm based that makes immediate use of the fact that colour and pattern information is readily available in the CVPIC domain. Colour features are exploited by building a block co-occurrence matrix of colour indices while shape information is represented through pattern histograms. Combining these two types of descriptors results in an efficient and effective image retrieval method that even outperforms popular pixel-based algorithms such as colour histograms, colour coherence vectors and colour correlograms.

1 Introduction

With the rise of the Internet and the availability of affordable digital imaging devices, the need for content-based image retrieval (CBIR) is ever increasing. While many methods have been suggested in the literature only few take into account the fact that - due to limited resources such as disk space and bandwidth - virtually all images are stored in compressed form. In order to process them for CBIR they first need to be uncompressed and the features calculated in the pixel domain. Often these features are stored along-

side the images which seems counterintuitive to the original need for compression. The desire for techniques that operate directly in the compressed domain providing, so-called midstream content access, seems therefore evident [6].

Colour Visual Pattern Image Coding (CVPIC) is one of the first so-called 4-th criterion image compression algorithms [9, 8]. A 4-th criterion algorithm allows - in addition to the classic three image coding criteria of image quality, efficiency, and bitrate - the image data to be queried and processed directly in its compressed form, in other words the image data is directly meaningful without the requirement of a decoding step. The data that is readily available in CVPIC compressed images is the colour information of each of the 4×4 blocks the image has been divided into, and information on the spatial characteristics of each block, in particular on whether a given block is identified as a uniform block (a block with no or little variation) or a pattern block (a block where an edge or gradient has been detected).

In this paper we make direct use of this information and propose an image retrieval algorithm that allows for efficient and effective retrieval directly in the compressed domain of CVPIC. We make use of the fact that colour and edge information is readily available in CVPIC. The colour features are used to build a block colour co-occurrence matrix similar as in [3] whereas the encoded edge information is summarised in a shape histogram. Comparing these histograms and integrating the two scores achieves image retrieval based on both (spatial) colour and shape. Experimental results obtained from querying the UCID [10] dataset show that this approach not only allows retrieval directly in the compressed domain but also outperforms techniques such as colour histograms, colour coherence vectors and color correlograms.

2 Colour Visual Pattern Image Coding

The Colour Visual Pattern Image Coding (CVPIC) image compression algorithm introduced by Schaefer et al. [9] is an extension of the work by Chen and Bovik [1]. The underlying idea is that within a 4×4 image block only one discontinuity is visually perceptible.

CVPIC first performs a conversion to the CIEL*a*b* colour space [2] as a more appropriate image representation. As many other colour spaces, CIEL*a*b* comprises one luminance and two chrominance channels; CIEL*a*b* however, was designed to be a uniform representation, meaning that equal differences in the colour space correspond to equal perceptual differences. A quantitative measurement of these colour differences was defined using the Euclidean distance in the L*a*b* space and is given in ΔE units.

A set of 14 patterns of 4×4 pixels has been defined in [1]. All these patterns contain one edge at various orientations (vertical, horizontal, plus and minus 45°) as can be seen in Figure 1 where + and - represent different intensities. In addition a uniform pattern where all intensities are equal is being used.

+++	+++	+++		---	---	---	
---	---	---	---	---	---	---	---
---	---	---	---	---	---	---	---
---	---	---	---	---	---	---	---
---	---	---	---	---	---	---	---
---	---	---	---	---	---	---	---
---	---	---	---	---	---	---	---
---	---	---	---	---	---	---	---

Figure 1. The 14 edge patterns used in CVPIC

The image is divided into 4×4 pixel blocks. Determining which visual pattern represents each block most accurately then follows. For each of the visual patterns the average L*a*b* values μ_+ and μ_- for the regions marked by + and - respectively (i.e. the mean values for the regions on each side of the pattern) are calculated.

The colour difference of each actual pixel and the corresponding mean value is obtained and averaged over the block according to

$$\epsilon = \frac{\sum_{i \in +} \|p_i - \mu_+\| + \sum_{j \in -} \|p_j - \mu_-\|}{16} \quad (1)$$

The visual pattern leading to the lowest ϵ value (given in CIEL*a*b* ΔE units) is then chosen. In order to allow for the encoding of uniform blocks the average colour difference to the mean colour of the block is also determined according to

$$\sigma = \frac{\sum_{\forall i} \|p_i - \mu\|}{16} \text{ where } \mu = \frac{\sum_{\forall i} p_i}{16} \quad (2)$$

A block is coded as uniform if either its variance in colour is very low, or if the resulting image quality will not suffer severely if it is coded as a uniform rather than as an edge block. To meet this requirement two thresholds

are defined. The first threshold describes the upper bound for variations within a block, i.e. the average colour difference to the mean colour of the block. Every block with a variance below this value will be encoded as uniform. The second threshold is related to the difference between the average colour variation within a block and the average colour difference that would result if the block were coded as a pattern block (i.e. the lowest variance possible for an edge block) which is calculated by

$$\delta = \sigma - \min_{\forall \text{patterns}}(\epsilon) \quad (3)$$

If this difference is very low (or if the variance for a uniform pattern is below those of all edge patterns in which case σ is negative) coding the block as uniform will not introduce distortions much more perceptible than if the block is coded as a pattern block. Hence, a block is coded as a uniform block if either σ or δ fall below the thresholds of $1.75 \Delta E$ and $1.25 \Delta E$ respectively (which we adopted from [9]).

For each block, one bit is stored which states whether the block is uniform or a pattern block. In addition, for edge blocks an index identifying the visual pattern needs to be stored. Following this procedure results in a representation of each block as 5 bits (1 + 4 as we use 14 patterns) for an edge block and 1 bit for a uniform block describing the spatial component, and the full colour information for one or two colours (for uniform and pattern blocks respectively).

In contrast to [9] where each image is colour quantised individually, the colour components are quantised to 64 universally pre-defined colours (we adopted those of [7]). Each colour can hence be encoded using 6 bits. Therefore, in total a uniform block takes 7 (= 1 + 6) bits, whereas a pattern block is stored in 17 (= 5 + 2 * 6) bits. We found that this yielded an average compression ratio of about 1:30. We note, that the information could be further encoded to achieve lower bitrates. Both the pattern and the colour information could be entropy coded. In this paper however, we refrain from this step as we are primarily interested in a synthesis of coding and retrieval.

3 CVPIC image retrieval

We note from above that for each image block in CVPIC both colour and edge information is readily available in the compressed form: each block contains either one or two colours and belongs to one of 15 edge classes (14 for edge blocks, and one for uniform blocks). We therefore propose to make direct use of this information for the purpose of image retrieval.

It is well known that colour is an important cue for image retrieval. In fact, simple descriptors such as histograms of the colour contents of images [12] have been shown to work well and have hence been used in many CBIR systems. Further improvements can be gained by incorporating spatial

information as techniques such as colour coherence vectors [5], border/interior pixel histograms [11], and colour correlograms [3] have shown. The latter builds histograms that record the probabilities that a pixel with a certain colour is a certain distance away from another pixel with another colour, formally expressed as

$$\gamma_{c_i, c_j}^{(k)}(I) = \Pr_{p_1 \in I_{c_i}, p_2 \in I} [p_2 \in I_{c_j}, |p_1 - p_2| = k] \quad (4)$$

with

$$|p_1 - p_2| = \max |x_1 - x_2|, |y_1 - y_2| \quad (5)$$

where c_i and c_j denote the two colours and (x_k, y_k) denote pixel locations. k is the set of distances that is considered and is often set to $k = \{1, 3, 5, 7\}$.

Our approach to exploiting the colour information that is readily available in CVPIC - block colour co-occurrence matrix - is similar to that of colour correlograms. However, in contrast to those it is very efficient to compute. We define the block colour co-occurrence matrix BCCM as

$$\text{BCCM}(c_i, c_j) = \Pr(p_1 = c_i, p_2 = c_j) \quad (6)$$

In other words for each CVPIC 4×4 subblock we increment the histogram bin that is defined by the two colours on each side of the edge in the block. We note, that for uniform blocks the bins along the 'diagonal' of the (64×64) diagonal histogram get incremented, i.e. those bins that essentially correspond to the auto correlogram $\alpha_c^{(k)}(I) = \gamma_{c,c}^{(k)}(I)$ [3]. Two BCCMs are compared by

$$d(I_1, I_2)_{\text{BCCM}} = \frac{\sum_{i=1}^{64} \sum_{j=i}^{64} |\text{BCCM}_1(i, j) - \text{BCCM}_2(i, j)|}{1 + \sum_{i=1}^{64} \sum_{j=i}^{64} \text{BCCM}_1(i, j) + \sum_{i=1}^{64} \sum_{j=i}^{64} \text{BCCM}_2(i, j)} \quad (7)$$

While image retrieval based on colour usually produces useful results, integration of this information with another paradigm such as texture or shape will result in an improved retrieval performance. Shape descriptors are often calculated as statistical summaries of local edge information such as in [4] where the edge orientation and magnitude is determined at each pixel location and an edge histogram calculated. Exploiting the CVPIC image structure an effective shape descriptor can be determined very efficiently. Since each (pattern) block contains exactly one (precalculated) edge and there are 14 different patterns we simply build a 1×14 histogram of the edge indices. We decided not to include a bin for uniform blocks, since these give little indication of shape (rather they describe the absence of it). Block edge histograms BEH_1 and BEH_2 are compared analogous to BCCMs by

$$d(I_1, I_2)_{\text{BEH}} = \frac{\sum_{k=1}^{14} |\text{BEH}_1(k) - \text{BEH}_2(k)|}{1 + \sum_{k=1}^{14} \text{BEH}_1(i, j) + \sum_{k=1}^{14} \text{BEH}_2(i, j)} \quad (8)$$

Having calculated the distances $d(I_1, I_2)_{\text{BCCM}}$ and $d(I_1, I_2)_{\text{BEH}}$ between two images I_1 and I_2 these two can be combined in order to allow for image retrieval based on both colour and shape features. Obviously, the simplest method of doing that would be to just add the two measures. While this will provide good enough results in many cases (as it did for the experiments we carried out in Section 4), depending on the database, better performance might be achieved by putting different weights on the two scores. Hence, the overall distance between I_1 and I_2 can be described by

$$d(I_1, I_2) = \alpha d(I_1, I_2)_{\text{BCCM}} + (1 - \alpha) d(I_1, I_2)_{\text{BEH}} \quad (9)$$

where a weighting factor $\alpha > 0.5$ would put more importance on the colour contents while $\alpha < 0.5$ puts more emphasis on the shape features.

4 Experimental Results

We evaluated our method using the recently released UCID dataset [10]. UCID, an Uncompressed Colour Image Database¹, consists of 1338 colour images all preserved in their uncompressed form which makes it ideal for the testing of compressed domain techniques. UCID also provides a ground truth of 262 assigned query images each with a number of predefined corresponding matches that an ideal image retrieval system would return.

We compressed the database using the CVPIC coding technique and performed image retrieval using the algorithm detailed in Section 3 (we set $\alpha = 0.5$ i.e. putting equal importance on colour and shape features) based on the queries defined in the UCID set. As performance measure we use the modified average match percentile (AMP) from [10] which is defined as

$$\text{MP}_Q = \frac{100}{S_Q} \sum_{i=1}^{S_Q} \frac{N - R_i}{N - i} \quad (10)$$

with $R_i < R_{i+1}$ and

$$\text{AMP} = \frac{1}{Q} \sum \text{MP}_Q \quad (11)$$

where R_i is the rank the i^{th} match to query image Q was returned, S_Q is the number of corresponding matches for Q , and N is the total number of images in the database. A perfect retrieval system would achieve an AMP of 100 whereas an AMP of 50 would mean the system performs as well as one that returns the images in random order.

In order to relate the results obtained we also implemented colour histogram based image retrieval [12], colour

¹UCID is available from <http://vision.doc.ntu.ac.uk/>.

	AMP
Colour histograms	90.47
Colour coherence vectors	91.03
Border/interior pixel histograms	91.27
Colour correlograms	89.96
CVPIC image retrieval	94.23

Table 1. Results obtained on the UCID dataset.

coherence vectors [5], border/interior pixel histograms [11] and colour auto correlograms [3]. Results for all methods can be found in Table 1. From there we see that our novel approach is not only capable of achieving good retrieval performance, but that it actually clearly outperforms all other methods! This is further illustrated in Figure 2 which shows one of the UCID query images together with the five top ranked images returned by all methods. Only the CVPIC techniques manages to retrieve five correct model images in the top five (with the next model coming up in sixth place) while colour correlograms retrieve three and all other methods only two. This is especially remarkable so as methods such as colour histograms, colour coherence vectors and colour correlograms are known to work fairly well for image retrieval and are hence among those techniques that are most widely used in this field.



Figure 2. Sample query together with 5 top ranked images returned by (from top to bottom) colour histograms, colour coherence vectors, border/interior pixel histograms, colour correlograms, CVPIC retrieval.

5 Conclusions

In this paper we present a novel image retrieval technique that operates directly in the compressed domain of CVPIC compressed images. Since colour and shape features are directly available in CVPIC encoded images we utilise these to compute a block co-occurrence matrix for colour features and a pattern histogram to summarise shape information. Experimental results on a medium-sized colour image database show that combined (spatial) colour and shape CVPIC retrieval performs extremely well, outperforming techniques such as colour histograms, colour coherence vectors, and colour correlograms.

Acknowledgements

The first author would like to acknowledge the support of the Nuffield Foundation under grant number NAL/00703/G.

References

- [1] D. Chen and A. Bovik. Visual pattern image coding. *IEEE Trans. Communications*, 38:2137–2146, 1990.
- [2] CIE. *Colorimetry*. CIE Publications 15.2, Commission International de L’Eclairage, 2nd edition, 1986.
- [3] J. Huang, S. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih. Image indexing using color correlograms. In *IEEE Int. Conference Computer Vision and Pattern Recognition*, pages 762–768, 1997.
- [4] A. Jain and A. Vailaya. Image retrieval using color and shape. *Pattern Recognition*, 29(8):1233–1244, 1996.
- [5] G. Pass and R. Zabih. Histogram refinement for content-based image retrieval. In *3rd IEEE Workshop on Applications of Computer Vision*, pages 96–102, 1996.
- [6] R. Picard. Content access for image/video coding: The fourth criterion. Technical Report 195, MIT Media Lab, 1994.
- [7] G. Qiu. Colour image indexing using BTC. *IEEE Trans. Image Processing*, 12(1):93–101, 2003.
- [8] G. Schaefer and G. Qiu. Midstream content access based on colour visual pattern coding. In *Storage and Retrieval for Image and Video Databases VIII*, volume 3972 of *Proceedings of SPIE*, pages 284–292, 2000.
- [9] G. Schaefer, G. Qiu, and M. Luo. Visual pattern based colour image compression. In *Visual Communication and Image Processing 1999*, volume 3653 of *Proceedings of SPIE*, pages 989–997, 1999.
- [10] G. Schaefer and M. Stich. UCID - An Uncompressed Colour Image Database. In *Storage and Retrieval Methods and Applications for Multimedia 2004*, volume 5307 of *Proceedings of SPIE*, pages 472–480, 2004.
- [11] R. Stehling, M. Nascimento, and A. Falcao. A compact and efficient image retrieval approach based on border/interior pixel classification. In *Proc. 11th Int. Conf. on Information and Knowledge Management*, pages 102–109, 2002.
- [12] M. Swain and D. Ballard. Color indexing. *Int. Journal Computer Vision*, 7(11):11–32, 1991.