Image retrieval at low bit rates: BSP Trees vs. JPEG

Michal Stich[1,2] and Gerald Schaefer[1]

[1]School of Computing and Technology
The Nottingham Trent University, Nottingham, U.K.
[2]Dept. of Computing, Electronics and Automated Control
Silesian University of Technology, Gliwice, Poland

KEYWORDS

Image retrieval, image compression, BSP tree, JPEG.

ABSTRACT

Content based image retrieval (CBIR) has still a long way to go before it will become capable of distinguishing similar rather than almost identical images. These limitations result both from a lack of 'intelligent' enough algorithms as well as problems in terms of computational restrictions which force algorithms to be as simple as possible, while maintaining reasonably high effectiveness. Images are not analysed in terms of their physical or logical contents in the way humans perceive it but most often as statistical information such as spatial correlation of colors or intensities of separate pixels. This information is usually extracted at a very local (pixel) level and may be miss-interpreted if pixel values change (even though the overall image will still appear similar), which is especially the case in image compression. Results of earlier experiments [10] show that this can cause notable problems for CBIR algorithms. In this paper, we try to address the problem by investigating a more appropriate compression algorithm based on binary space partitioning trees and how it can improve the retrieval performance of compressed images. Simulations on two different image databases show that BSP compression outperforms JPEG for colour-based image retrieval while for texture-based indexing JPEG performs better.

INTRODUCTION

In our previous work [10] we have studied the influence of standard JPEG compression on the effectiveness of several image retrieval approaches: color histograms, QBIC histograms, colour moments, colour correlograms, spatial-chromatic histograms and color coherence vectors. Results of these experiments showed that while slight compression has little effect on colour based CBIR it plays a significant role when higher compression levels are applied resulting in a notably reduced image retrieval performance.

In this paper we try to present one possible solution to this problem by evaluating an alternative compression approach – binary space partitioning (BSP) trees [9,8]. This compression method we compared to the most commonly used image coding algorithm: JPEG [14]. Our experimental imagery set is UCID, an Uncompressed Color Image Database [11] which is the same dataset used in [10]. In the experiments we compress these images to very low bitrates using JPEG and to a level giving slightly lower bit rate using BSP. Our results show that in contrast to JPEG image compression based on binary space partitioning can be used in color image retrieval for low bit-rates almost without any loss in performance. On the other hand, texture retrieval results obtained from the Brodatz texture set shows a significant drop in retrieval performance.

The rest of the paper is organised as follows: the next section describes the BSP compression algorithm used in the experiments. Then the colour and texture retrieval algorithms evaluated are briefly explained. The following section presents our experimental results while finally concluding remarks are presented.

BSP TREE COMPRESSION

Binary Space Partitioning (BSP) is a relatively new approach to image compression. It was originally used to represent a three dimensional space for convenience of hidden surface removal algorithms and was recently successfully applied to 2D imagery as well [9,8].

The idea is to divide an image plane by one of several pre-defined straight lines. This process is recursively repeated for each of the two sub-regions created by the previous partition. At some point partitioning is stopped and basic information about the fraction of the entire space is stored. The coded information is organised in a tree structure where each node indicates a part of space and contains information about the partitioning line, while its two children point to two sub-paces created as a result of the partitioning of their parent. The bottom-most nodes of the tree contain information about that region (e.g. color of this region of the image).

The first attempt to compress images using BSP was made by Radha *et al.* [9] whose algorithm utilised the idea of moment preserving thresholding and was only applied to gray-scale images. Qiu and Sudirman extended this idea to colour images [8]. In our implementation, rather than using moment preserving thresholding we calculate the average color of the

Proceedings 18th European Simulation Multiconference
Graham Horton (c) SCS Europe, 2004
ISBN 3-936150-35-4 (book) / ISBN 3-936150-36-2 (CD)

partitioned regions and then the resulting errors as the sum of differences between intensities for each pixel in the region and the average intensity of region to which this pixel belongs. The partitioning line is then chosen so as to minimize the error. While this approach is clearly computationally more expensive it has the advantage of 'optimality' and hence better image quality. We also employ the CIEL*a*b* color space [2] which gives improved quality in comparison to RGB.

CBIR ALGORITHMS

In this section we provide a brief description of the colour and texture CBIR algorithms that were used for the experiments.

Colour Histograms – Histogram Intersection

Given a bounded, discrete signal one can build a histogram simply by counting the number of occurrences of each signal value. Swain and Ballard [13] were the first to use colour histograms to describe images in order to perform object recognition and image retrieval. Indeed, it was Swain and Ballard's work that laid the foundations for the field of CBIR as we know it today. As distance measure they introduced (the complement of) histogram intersection defined as

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

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. It can be shown [13] that histogram intersection is equivalent to the L_1 norm and hence a metric. We used 8 x 8 x 8 RGB histograms in our experiments.

Colour Histograms - QBIC

An alternative to the L_1 norm is to use the Euclidean distance (L_2 norm) between two histograms. This approach was taken in the QBIC system [4] where they also addressed 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

$$d_{\text{QBIC}}(I_1, I_2) = (H_1 - H_2)A(H_1 - H_2)^T \quad (2)$$

where H_1 and H_2 are again the two colour histograms (in the form of a vector) and A is an N x N matrix containing the inter-bin distances. We used the Munsell colour space divided into 256 bins (16 for hue, 4 for chroma and value respectively) to generate these histograms.

Colour Moments

Stricker and Orengo [12] used colour moments as a compact colour descriptor for CBIR. The n^{th} central (normalised) moment of a colour distribution is defined as

$$M^n(I) = \sqrt[n]{\frac{1}{N} \sum \left(M^1(I) - c(x,y)\right)^n} \quad (3)$$

with

$$M^1(I) = \frac{1}{N} \sum c(x,y) \quad (4)$$

where N is the number of pixels in an image and $c(x,y)$ describes the colour of the pixel at location (x,y). For our experiments we used the first three moments in the HSV colour space. The distance between two images is defined as the sum of absolute distances between their moments (L_1 norm)

$$d_{\text{MNT}}(I_1, I_2) = \sum_{i=1}^{n} \left| M^i(I_1) - M^i(I_2) \right| \quad (5)$$

Color Coherence Vectors.

Pass and Zabih [7] introduced colour coherence vectors as a method of integrating 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 τ where τ is usually defined as 1% of the overall area of an image. The L_1 norm is used as the distance metric between two colour coherence vectors

$$d_{\text{CCV}}(I_1, I_2) = \sum_{k=1}^{N} \begin{bmatrix} \left| H_1^c(k) - H_2^c(k) \right| + \\ \left| H_1^s(k) - H_2^s(k) \right| \end{bmatrix} \quad (6)$$

where H_i^c and H_i^s are the histograms of coherent and non-coherent (scattered) pixels respectively. In our implementation we first blurred the image using a 3 x 3 averaging filter and used 8 x 8 x 8 RGB bins for representing the histograms.

Colour Correlograms

Another approach to incorporate information on the spatial correlation between the colours present in an image was proposed by Huang *et al.* [5]. They introduced the notation of colour correlograms (CCRs) defined as

$$\gamma_{c_i c_j}^{(k)}(I) \equiv \Pr_{p_1 \in I_{c_i}, p_2 \in I} \left[p_2 \in I_{c_j}, \left| p_1 - p_2 \right| = k \right] \quad (7)$$

with

$$\left| p_1 - p_2 \right| = \max\left\{ \left| x_1 - x_2 \right|, \left| y_1 - y_2 \right| \right\} \quad (8)$$

where c_i and c_j denote two colours and (x_k, y_k) denote pixel locations. In other words, given any colour c_i in

the image, γ gives the probability that a pixel at distance k away is of colour c_j.
As full colour correlograms are expensive both in terms of computation and storage requirements, usually a simpler form called auto-correlogram (ACR) defined as

$$\alpha_c^{(k)} \equiv \gamma_{c,c}^{(k)}(I) \quad (9)$$

is being used, i.e. only the spatial correlation of each colour to itself is recorded. Two CCRs are compared using

$$d_{\mathrm{CCR}}(I_1, I_2) = \frac{\sum_{i,j\in[m],k\in[d]} \left|\gamma_{c_i,c_j}^{(k)}(I) - \gamma_{c_i,c_j}^{(k)}(I')\right|}{\sum_{i,j\in[m],k\in[d]} \left(1 + \gamma_{c_i,c_j}^{(k)}(I) + \gamma_{c_i,c_j}^{(k)}(I')\right)} \quad (10)$$

We used ACRs with 8 x 8 x 8 RGB colours, for k we chose {1,3,5,7}.

Spatial–chromatic Histograms

Cinque *et al.* [3] introduced spatial-chromatic histograms (SCHs) as an alternative method for representing both colour and spatial information. SCHs consist of a colour histogram

$$h(k) = \frac{|A_k|}{n * m} \quad (11)$$

where A_k is a set having the same colour k, and n and m are the dimensions of the image;
and location information on each colour characterised through its baricentre

$$b(k) \equiv \left(\frac{1}{n}\frac{1}{|A_k|}\sum_{(x,y)\in A_k} x ; \frac{1}{m}\frac{1}{|A_k|}\sum_{(x,y)\in A_k} y\right) \quad (12)$$

and the standard deviation of distances of a given colour from its baricentre

$$\sigma(k) = \sqrt{\frac{1}{|A_k|}\sum_{p\in A_k} d(p, b(k))^2} \quad (13)$$

The SCH is then given as

$$H_{\mathrm{SCH}}(k) = [h(k), b(k), \sigma(k)] \quad (14)$$

and similarity between two SCHs calculated as

$$d_{\mathrm{SCH}}(I_1, I_2) = 2 - \sum_{i=1}^{N} \min\left(h_{I_1}(i), h_{I_2}(i)\right)\left(\frac{\sqrt{2} - d\left(b_{I_1}(i), b_{I_2}(i)\right)}{\sqrt{2}} + \frac{\min\left(\sigma_{I_1}(i), \sigma_{I_2}(i)\right)}{\max\left(\sigma_{I_1}(i), \sigma_{I_2}(i)\right)}\right) \quad (15)$$

In our implementation we divided the Munsell colour space uniformly into 512 areas whose centres were used as the colours to describe the SCH.

3.7. Local Binary Patterns

Local Binary Patterns (LBP) [6] is a very simple method for texture retrieval. For each pixel in the image and its 8 neighbours the following operation is performed:

$$LBP = \sum_{i=1}^{8} T * W_i \quad (16)$$

with

$$T = 1 \text{ if } p_i >= P$$
$$T = 0 \text{ if } p_i < P \quad (17)$$

where P is the intensity of the centre pixel, p_i the intensity of its i^{th} neighbor and W_i is the weight (a power of 2) associated with p_i. The idea is to create histograms of 256 possible values (since each pixel has 8 neighbours). Two LBP histograms H_1^{LBP} and H_2^{LBP} can be then compared by the histogram intersection

$$d_{\mathrm{LBP}}(I_1, I_2) = 1 - \sum_{k=1}^{256} \min(H_1^{LBP}(k), H_2^{LBP}(k)) \quad (18)$$

Rotation invariant LBP

A rotation invariant version of the LBP algorithm can be deduced by finding all groups of patterns that can be obtained through rotation of the 8 neighbouring pixels [6] by

$$riLBP(k) = \sum_{i=1}^{8} T_i * W_i^{(k)} \quad (19)$$

and then choosing the smallest LBP value

$$riLBP' = \min(riLBP(k)), k = 1,2,...,8 \quad (20)$$

which results in 36 possible descriptors that are again summarized in histograms. Two rotation invariant LBP histograms H_1^{riLBP} and H_2^{riLBP} are compared by

$$d_{\mathrm{riLBP}}(I_1, I_2) = 1 - \sum_{k=1}^{36} \min(H_1^{riLBP}(k), H_2^{riLBP}(k)) \quad (21)$$

EXPERIMENTAL RESULTS

For our experiments we adopted the UCID (Uncompressed Colour Image Database) [11] for evaluating the colour descriptors while we used Brodatz textures [1] for performing texture based image retrieval. The UCID set consists of 1338 colour images all preserved in their uncompressed state. It also provides a ground truth of 262 assigned query images each together with a set of corresponding model images that an ideal image retrieval system would return. The

112 Brodatz textures we divided into 9 non-overlapping parts each and assigned the centre block the query and the rest the model images that should be retrieved.

As performance measure we use the retrieval effectiveness from [4] which is given by

$$\mathrm{RE}_Q = \sum_{i=1}^{S_Q} R_i \Big/ \sum_{i=1}^{S_Q} I_i \quad (21)$$

where R_i is the rank of the i^{th} matching image and I_i is the ideal rank of the i^{th} match (i.e. $I = \{1,2,\ldots,S_Q\}$). The average retrieval effectiveness ARE is then taken as the mean of RE over all query images.

For the experiments both query and model images were heavily compressed using the JPEG algorithm (q-factor of 5) and to a level giving similar (although even slightly lower) bit rate using BSP compression. Initial observations lead to the conclusion, that low bit rate BSP compression retains global colour information of the image, which is not the case for JPEG – heavily compressed JPEG images tend to almost entirely loose their colour content while the contours are still visible (though heavily affected by blocking effects). Conversely, BSP images keep their colours, while shapes are preserved as clearly.

Table1: Image retrieval results from the UCID dataset.

CBIR algorithm	Q orig M orig	JPEG Q cmp M orig	JPEG Q orig M cmp	BSP Q cmp M orig	BSP Q orig M cmp
Colour hist.	4.31	25.73	13.48	4.71	4.63
QBIC	5.64	23.86	19.56	6.75	7.47
Colour moments	7.33	27.59	33.93	7.86	11.64
Colour coh. vect.	4.69	22.00	13.97	4.65	5.35
Colour correlogr.	5.32	11.80	8.91	5.69	5.97
Spat.-chrom. hist.	3.69	23.88	12.49	4.25	3.80

Figure 1: Example UCID query together with top 9 retrieved (using spatial-chromatic histograms) images both based on JPEG (left) and BSP (right) encoded images.

We performed image retrieval on the datasets mentioned above, first by using only uncompressed images in order to get a 'best possible' retrieval performance which we can use as a benchmark. We then retrieved the images once with the model images compressed and once with the query images compressed to the levels indicated above, both for JPEG and for BSP compression. The results from the UCID dataset are listed in Table 1. It is obvious that our previous expectations are confirmed if not surpassed: while image retrieval performance at low bitrates drops significantly for JPEG compression, image retrieval based on BSP is almost independent of compression. Retrieval performance is still very good even at these high compression ratios of well over 1:100! An example of this performance difference can be seen in Figure 1 where a UCID query image is shown together with the top nine retrieved images, both for JPEG and BSP encoded images. While the BSP based retrieval retrieves all three model images in top ranks, none of the JPEG models are returned. Also, we can see that the images retrieved in ranks 4-9 are much more useful (three taken at the same location) than those returned by JPEG compressed images.

Table 1. Image retrieval results from the Brodatz dataset.

CBIR algorithm	Q orig M orig	JPEG Q cmp M orig	JPEG Q orig M cmp	BSP Q cmp M orig	BSP Q orig M cmp
LBP	3.60	64.40	33.24	80.12	71.03
Rot.inv. LBP	5.05	81.69	57.94	95.45	92.29

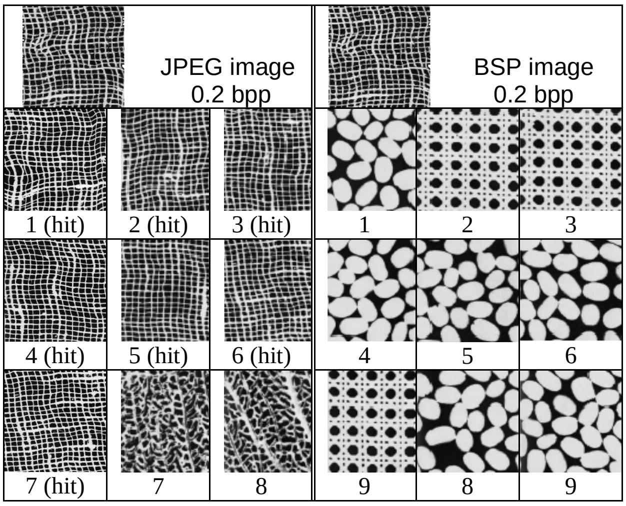

Figure 2: Example UCID query together with top 9 retrieved (using LBP) images both based on JPEG (left) and BSP (right) encoded images.

Results for texture retrieval based on the Brodatz set show a different picture. Since BSP essentially divides an image in areas of uniform colour pixel-based statistics such as LBP and riLBP get heavily distorted. The image retrieval results are given in Table 2 from where we can see that while JPEG based image retrieval suffers from a significant performance drop this drop is even greater so for BSP images. Again we provide a visual example of this which is given in Figure 2.

CONCLUSIONS

Through an extensive set of simulations on two different image databases we have demonstrated that different compression algorithms will affect the outcome of image retrieval differently at high compression. While JPEG suffers from a significant performance drop for colour based CBIR, retrieval of BSP compressed images is almost independent of the compression rate. Therefore we suggest BSP as a much more appropriate compression method that can be used for image retrieval even at very low bitrates. On the other hand, texture based retrieval results are much less favourable, here BSP performs even worse than JPEG. Finally we want to emphasise that in this paper we made use only of the pixel based representations of the compressed images (hence, essentially decompressing them again). A more promising route is to couple the retrieval algorithm with the compression hence achieving compressed domain image retrieval. This alternative is not only attractive in terms of computational complexity but seems also to provide very good retrieval results as some initial work in [8] shows.

REFERENCES

[1] P. Brodatz. "Textures", Dover, New York, 1966.

[2] CIE "Colorimetry", 2nd edition, *CIE Publ. No. 15.2*, Commission International de L'Eclairage, Vienna, 1986.

[3] L. Cinque, S. Levialdi, A. Pellicano. "Color-Based Image Retrieval using Spatial-Chromatic Histograms." In *Proc. IEEE Multimedia Systems 99,* pp. 969-973, 1999.

[4] C. Faloutsos, W. Equitz, M. Flickner, W. Niblack, D. Petkovic, R. Barber. "Efficient and Effective Querying by Image Content." *Journal of Intelligent Information Systems,* Vol.3, No.3/4, pp.231-262, 1994.

[5] J. Huang, S. R. Kumar, M. Mitra, Wei-Jing Zhu, R. Zabih. "Image Indexing Using Color Correlograms." In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 762-768, 1997.

[6] T. Ojala, M. Pietikäinen, T. Mäenpää, "Gray Scale Rotation Invariant Texture Classification with Local Binary Patterns." *Proc. European Conference on Cumpter* Vision, 2000.

[7] G. Pass, R. Zabih. "Histogram Refinement for Content-Based Image Retrieval." In *IEEE Workshop on Applications of Computer Vision*, pp. 96-102, 1996.

[8] G. Qiu, S. Sudirman "Color Image Coding, Indexing and Retrieval using Binary Space Partitioning Tree." In *Proceeding to 8th Color Image Conference*, pages 195-201, 2000.

[9] H. Radha, et al "Image compression using binary space partitioning tree." *IEEE Trans. On Image Processing*, vol.5, pages 1610-1624, 1996.

[10] G. Schaefer, M. Stich "On the influence of image compression on the performance of content based image retrieval." In *6th International Conference on VISual Information Systems*, 2003.

[11] G. Schaefer, M. Stich. "UCID – An Uncompressed Colour Image Database." In *Storage and Retrieval Methods and Applications for Multimedia* 2004, 2004.

[12] M. A. Stricker, M. Orengo. "Similarity of Color Images." In *Storage and Retrieval for Image and Video Databases III*, pp. 381-392. 1995.

[13] M. Swain, D. Ballard. "Color Indexing." *International Journal of Computer Vision*, 7:1, pp. 11-32. 1991.

[14] G.K. Wallace. "The JPEG Still Picture Compression Standard." *Communications of the ACM*, Vol. 34, No. 4, pp. 30-44, 1991.