

Compressed Domain Image Retrieval Using JPEG2000 and Gaussian Mixture Models

Alexandra Teynor¹, Wolfgang Müller², and Wolfgang Kowarschick³

¹ Albert-Ludwigs-University of Freiburg, Institute of Pattern Recognition and Image Processing, 79110 Freiburg, Germany teynor@informatik.uni-freiburg.de

² Bamberg University, LS Medieninformatik, 96045 Bamberg, Germany
Wolfgang.Mueller@wiai.uni-bamberg.de

³ Augsburg University of Applied Sciences, Department of Computer Science, 86169 Augsburg, Germany kowa@fh-augsburg.de

Abstract. We describe and compare three probabilistic ways to perform Content Based Image Retrieval (CBIR) in compressed domain using images in JPEG2000 format. Our main focus are arbitrary non-uniformly textured color images, as can be found, e.g., in home user image collections. JPEG2000 offers data that can be easily transferred into features for image retrieval. Thus, when converting images to JPEG2000, feature extraction comes at a low cost. For feature creation, wavelet subband data is used. Color and texture features are modelled independently and can be weighted by the user in the retrieval process. For texture features in common databases, we show in which cases modelling wavelet coefficient distributions with Gaussian Mixture Models (GMM) is superior in to approaches with Generalized Gaussian Densities (GGD). Empirical tests with data collected by non expert users evaluate the usefulness of the ideas presented.

1 Introduction

Content Based Image Retrieval (CBIR) has been motivated many times. While there are currently great advances in semantic features and image annotations, the proliferation of digital cameras operated by annotation-lazy novices motivates the automatic, feature based CBIR today. JPEG2000 [2] compressed domain CBIR draws its interest from the fact that probabilistic texture modelling techniques have been developed based on wavelets.

In this paper, we show how to compute and compare color and texture features based on wavelet subband data provided by JPEG2000. The main focus of this work is to show how subband data can be modelled accurately in order to capture the properties of non-uniformly textured images better. Work already done in this area mainly focuses on uniform texture images [9][16], or homogeneous areas in general images [4]. However, especially personal image collections do not consist of pure texture images, nor are users always interested in only partial matches. An other way to use JPEG2000 data for image retrieval was described by Mandal et al. [12]. They propose two JPEG2000 indexing techniques

based on significant bitmaps and the properties of bitplanes used to encode individual code blocks, however using subband data directly offers more information.

The outline of this paper is as follows: Section two deals with color and texture features derived from subband data of JPEG2000 images. In section three, evaluation methods are described and results are shown. Section four concludes this work.

2 Features from JPEG2000 data

Minka and Picard state in [13] that searching for the ultimate feature that represents an image is usually not successful, instead, it is better to rely on a number of primitive features and to combine them. We follow this approach. For arbitrary image databases of natural scenes, color and texture features are considered most important. We compute and compare them independently, getting two individual rankings for the best matches. To get the final result, the scores of each image in both rankings are combined. The default is to consider color and texture equally important, but the user has the possibility to emphasize either part by specifying a weight factor.

2.1 Data for computing features

The JPEG2000 image compression standard is rather complex and consists of different steps, details can be found, e.g., in [2]. During en-/decoding, there exists a state where the data is especially suited for computing features for CBIR. We extract features after the three color layers YC_bC_r have been wavelet transformed (each layer independently). Each color layer is represented by a number of subbands, depending on the number of wavelet decomposition steps. Layer Y carries the luminance information of the image. It can be interpreted as a black and white representation of the original image. Layers C_b and C_r account for chrominance. A graphical representation of the wavelet coefficients from a sample layer Y with seven subbands in different orientations is shown in figure 1.

For computing features, a JPEG2000 image has to be partially decompressed until it is in the format described above. A more efficient way would be to perform feature extraction right at compression time.

2.2 Color Features

On the lowest resolution subbands (LL_0) of layers Y , C_b and C_r , only low pass filtering and down sampling was performed during wavelet transform. Hence we can interpret these three subbands as a down scaled version of the original image in YC_bC_r space. This data can be used to create a color histogram [15] by assigning the color values to 5x8x8 bins. The Y -component has a coarser binning, since we want to be robust against small illumination changes. A similar method for creating color features for JPEG2000 images was proposed by [4], however

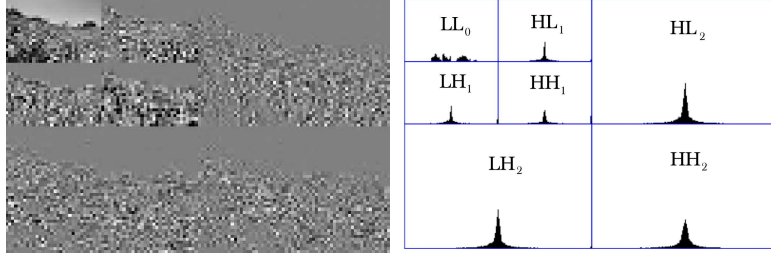


Fig. 1. Wavelet coefficients and their distribution for different subbands

they used more and equal sized bins for all color layers. In order to find out the color similarity of two images I_1 and I_2 , the histograms are compared using the well-known histogram intersection (HI).

2.3 Texture Features with Generalized Gaussian Densities

The higher order subbands represent horizontally and/or vertically high pass filtered data and therefore the “details” of the image. As shown in [9], the distribution of wavelet coefficients belonging to different subbands can represent the texture of images. In figure 1 histograms of subband data of the example image can be seen. For texture analysis, it is sufficient to consider only subbands of layer Y , being a black and white version of the image.

Do and Vetterli proposed in [9] to model the distribution of wavelet coefficients by a Generalized Gaussian Density (GGD):

$$p(x; \alpha, \beta) = \frac{\beta}{2\alpha\Gamma(\frac{1}{\beta})} e^{-(|x|/\alpha)^\beta} \quad (1)$$

where $\Gamma(\cdot)$ is the Gamma function: $\Gamma(z) = \int_0^\infty e^{-t} t^{z-1} dt$, $z > 0$

Here only two parameters α and β are necessary to describe the distribution. This is a very compact representation compared to, e.g., the histogram, where a large number of bins would be necessary to model the distribution satisfyingly. As described in [9], a way to estimate the parameters α and β in a statistical framework is by a maximum-likelihood (ML) estimator, which can be computed efficiently.

The similarity of two GGD can be determined using the Kullback-Leibler divergence or relative entropy [9]. In general terms, the KLD between two probability density functions (PDFs) $p_1(X)$ and $p_2(X)$ is defined as:

$$KLD[p_1(X)||p_2(X)] = \int p_1(x) \log \frac{p_1(x)}{p_2(x)} dx \quad (2)$$

For the KLD on GGDs there exists a closed form expression, where only the model parameters α and β are involved [9].

$$\begin{aligned}
KLD[p(X; \alpha_1, \beta_1) || p(X; \alpha_2, \beta_2)] = & \log \left(\frac{\beta_1 \alpha_2 \Gamma(1/\beta_2)}{\beta_2 \alpha_1 \Gamma(1/\beta_1)} \right) \\
& + \left(\frac{\alpha_1}{\alpha_2} \right)^{\beta_2} \frac{\Gamma((\beta_2 + 1)/\beta_1)}{\Gamma(1/\beta_1)} \\
& - \frac{1}{\beta_1}
\end{aligned} \tag{3}$$

Assuming that the wavelet coefficients in different subbands are independent, we can compute the overall texture similarity by applying the chain rule [9]. This means, in order to compute the overall KLD between images I_1 and I_2 , we can simply sum the KLD between corresponding subbands:

$$KLD[I_1 || I_2] = \sum_{j=1}^N KL[p_1^j(X) || p_2^j(X)] \tag{4}$$

with p_i^j denoting the PDF of the j -th subband of image i and N the number of subbands. In this work we assume that the number of decomposition levels for all images in the entire image database is the same, since the different subbands are matched one to another.

We examined the fitting quality of GGDs to real wavelet coefficient distributions in different cases. We performed tests with data from very coarse images, very smooth images and images that contain either part. For uniformly textured images, the fidelity is usually well, however it is not sufficient for images that have distinct regions. Especially affected are images that contain hard partitioned smooth and coarse regions, e.g., a picture of a landscape with a sky, or an object in front of a uniform background. However, these types of images are common in standard user image databases. Figure 2 shows sample images for three test cases, each with a wavelet coefficient distribution of a characteristic subband next to it. The dashed blue curves show how well the GGDs model the subband data distribution, respectively. The problem becomes evident when looking at the image with distinct regions. The wavelet coefficients belonging to the sky are close to zero and are forming a peak, the ones representing the crowd are very wide spread and form a kind of base. The GGD can not model this kind of distribution accurately.

2.4 Wavelet coefficient distribution with Gaussian Mixture Models

Because of this fitting inaccuracy, we propose to model the wavelet coefficient distribution as Gaussian Mixture Model (GMM). A similar proposition was also made by Crouse et al. [6] who use a two state zero mean Gaussian mixture model to build a hidden Markov model for characterizing wavelet transformed signals. Do et al. [8] also used these hidden Markov models for texture description. In our case, we do not use hidden Markov models, but a single GMM for each subband directly.

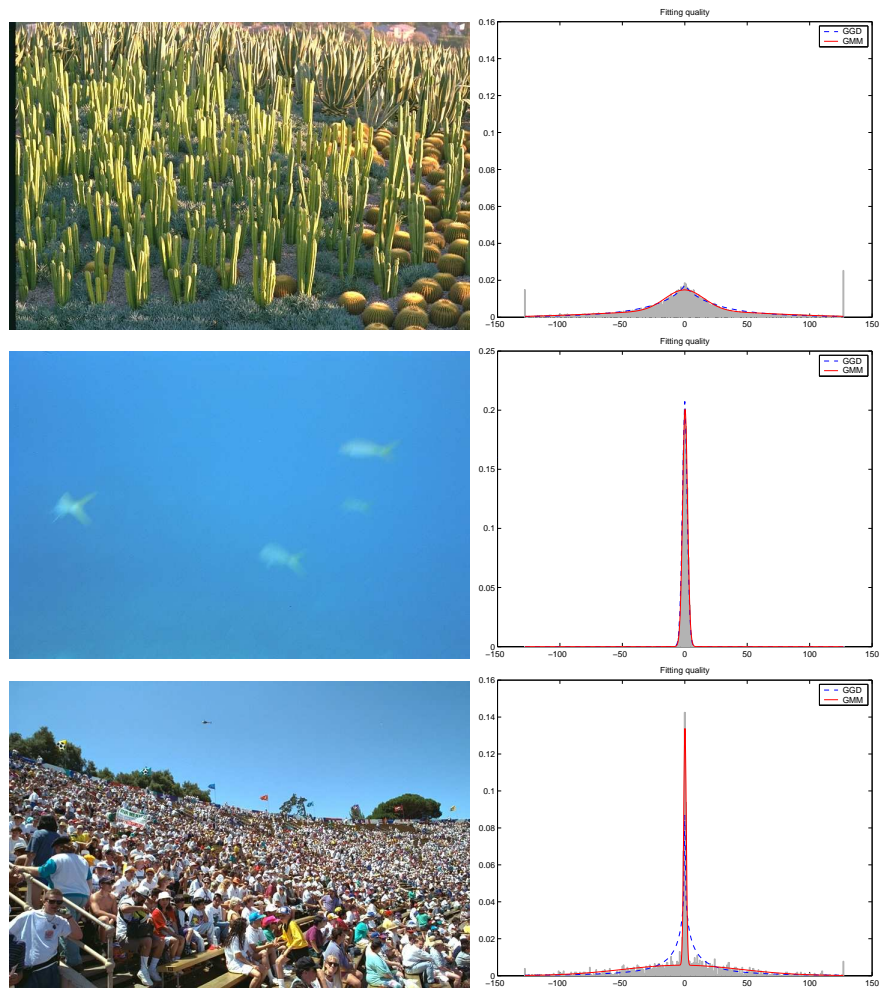


Fig. 2. Wavelet coefficient distribution of sample subbands, fitted with GGD and GMM with 2 components. Images are from the Benchathlon collection [1].

A mixture density has the form

$$p(x; \Theta) = \sum_{i=1}^N w_i p_i(x; \theta_i) \quad (5)$$

where the parameters are $\Theta = (w_1, \dots, w_N; \theta_1, \dots, \theta_N)$, such that $\sum_{i=1}^N w_i = 1$, and each p_i is a density function parameterized by θ_i . The functions used can be any valid probability density function, i.e. any set of non-negative functions integrating to one. In the case of Gaussian mixture models, Gaussian probability density functions are used.

The parameters w_i and θ_i (here μ_i and σ_i) can be estimated using the Expectation-Maximization (EM) algorithm [5]. It is an iterative procedure that needs good starting values and the number of mixture components necessary to model the distribution accurately. In a general setting this is a problem. But we already have knowledge how the mixture model to be estimated looks like. The wavelet coefficient distribution of a subband of a natural image is centered about zero and approximately symmetric (see figure 2). This means a small number of Gaussian mixtures with mean $\mu \approx 0$ will be sufficient. Experiments showed that already GMMs with two components model problematic wavelet coefficient distributions better than GGDs.

In figure 2, the fitting quality of GMMs for different subband data can be viewed (solid red curves). In this experiment, we used GMMs with two components. In the first two cases, where the GGD already performed well, the fidelity of the GMM is comparable. In the third case, modelling the subband data with GMMs is much more accurate compared to using GGDs.

In contrast to GGDs, there does not exist a simple way to compute the KLD on GMMs. A general approach is to use a Monte-Carlo (MC) estimation to approximate the integral in equation 2 [7]. Then, the KLD between the mixture densities $p_1(X)$ and $p_2(X)$ is estimated by:

$$KLD[p_1(X)||p_2(X)] \approx \frac{1}{N_{rnd}} \sum_{i=1}^{N_{rnd}} \log \frac{p_1(x_i)}{p_2(x_i)} \quad (6)$$

where the sample values $\{x_1, \dots, x_{N_{rnd}}\}$ are drawn randomly and independently from the model density $p_1(X)$. The number of N_{rnd} has usually to be large which spoils retrieval time, however, we get improved results. Experiments showed that 60 to 100 samples are sufficient.

Several scientists [7, 11, 10] have proposed ideas to compute the similarity between GMMs more efficiently. We used the approach described by Goldberger et al. [10]. They state different approximations of the KLD between mixture densities. As our GMMs have the same number of components, we used the method for calculating an upper bound for the KLD on mixture models. Details on this can be found in [10]. In the following, we refer to this method as ‘‘GMM-UB’’.

3 Experimental Evaluation

In order to compare the efficiency of the methods described above, we implemented a test framework. As image database we used a subset of the Benchathlon collection [1] (4501 images) which consists of consumer photos. Since there is no official ground truth available, we collected them by ourselves. We designed a tool for acquiring ground truth by non expert users⁴. The test persons were shown a query image and had to look through the whole database in order to look for similar images. The subjects had no previous experience with image retrieval systems. They were told to select images that they wanted to have retrieved by a system when presented the query image. So this ground truth can be considered quite hard, since not only visually similar images were selected, but also semantically similar ones. We acquired ground truth for a set of 11 randomly selected images by three different test persons, since similarity tends to be judged differently by different persons.

For evaluation we generated precision/recall plots, averaged over all three test persons and query images. Precision and recall are defined as [14]:

$$precision = \frac{|\mathbf{R} \cap \mathbf{T}|}{|\mathbf{T}|} \quad (7)$$

$$recall = \frac{|\mathbf{R} \cap \mathbf{T}|}{|\mathbf{R}|} \quad (8)$$

where \mathbf{R} is the set of images that are relevant to the query, \mathbf{T} is the set of returned images and $|\mathbf{A}|$ is the cardinality of set \mathbf{A} .

3.1 Evaluation of Features and Measures

The procedure for evaluating the algorithms was as follows: As explained in section 2, we calculated two different feature sets, one for color and one for texture. In order to access the wavelet coefficients in the JPEG2000 images, we used a modified version of the JasPer JPEG2000 codec [3]. Separate queries for color and texture similarity were performed implicitly and the individual rankings combined on the basis of the score of each image. As score of an image, a weighted sum of the normalized texture and color similarity measure is used. For color, this is the normalized HI value, for texture it is $1/(1 - KLD)$. These values are always between 0 and 1. The more similar an image is regarded by the system, the closer its value is to 1, with the query image itself having the maximum score of 1. As we evaluate arbitrary color pictures, equal weighting of the different feature types was performed. The texture features and the way their similarity is determined change in every experiment, color features however stay the same and thus will not be mentioned explicitly. When calculating the graphs for the MC method, the retrieval process was performed 3 times and the results averaged, due to the random nature of MC integration and the therefore slightly variable results.

⁴ This tool is available under http://muscle.prip.tuwien.ac.at/software_here.php

GGD



GMM-MC-60



GMM-UB



Fig. 3. Example query results. The query image is in the upper left corner.

As could be seen in figure 2, the modelling of the data only improved for images with hard partitioned textured and smooth areas. So a significant improvement is only expected here. For this reason, we evaluate the results for the “crowd and sky” image independently from the other images in the ground truth. Figure 3 shows a screen shot of the 10 best matches for images with texture modelled with GGDs and GMMs respectively. In the GMM case, the matching was performed with Monte Carlo integration with 60 samples (GMM-MC60). As we can see, the results improved noticeable.

A more precise evaluation can be done by looking at the precision/recall plots. In figure 4 on the left, the retrieval results for the different experiments are displayed. In this example, the GMM-based queries (GMM-UB and GMM-MC60) performed better than the GGD based one, since the precision is higher for the same level of recall. The best result was achieved by the GMM-MC method, because it has the highest recall values at the beginning of the precision/recall plot. This means that after few images retrieved, more relevant images are found compared to the other methods (the user wants to browse as few images as possible).

The average result for all images in the ground truth can be viewed in figure 4 on the right. Here also the advantage of the GMM method can be seen, however not that clearly, since many of the randomly selected query images do not have a problematic distribution.

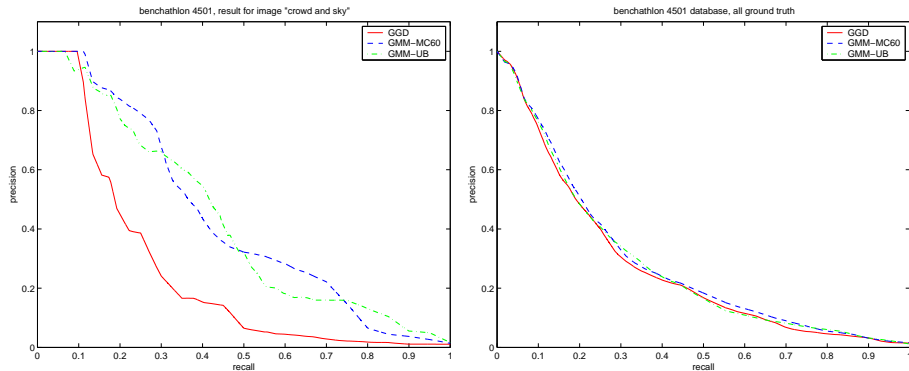


Fig. 4. Results for the problematic image and the entire ground truth

The resources needed by the different algorithms vary. In the GGD case, two parameters (α, β) describe the wavelet coefficient distribution per subband, in the GMM case we need $3 \times N$ with N being the number of mixture components (2 in our case). Since the JPEG2000 images used had 5 decomposition levels, the number of coefficients to characterize texture information in the image is $12 \times 2 = 24$ for GGDs and $12 \times 6 = 72$ for GMMs (we have 4×3 high pass subbands).

More severe are the differences in matching time for an image pair: for the calculation of the KLD on GGDs a single expression needs to be evaluated (see equation 3), so this is rather quick. When using the MC approximation, the retrieval time depends linearly on N_{rnd} , the number of random samples drawn. The calculation speed of the KLD for GMMs with an upper bound depends on the number of mixture components N . As described in [10], in order to obtain a tight upper bound for mixture models with equal N , minimizing over $N!$ permutations of the mixing components is necessary. However, since we use only $N = 2$ Gaussians, this is much less than the MC integration.

The absolute retrieval times for a texture query in our java framework on a P4 2.8GHz machine for 4501 images were about 600 ms for the GGD model, 14000 ms for GMM model with MC computation with 60 samples and 900 ms for the GMM-UB method. The evaluation time for the color features is the same in all cases. There the query time depends on the number of non-empty bins to be compared for histogram intersection, on average this was about 700 ms. We see, the computation of the KLD on GMMs with an upper bound is only 50% slower as computation of GGD similarity, while the method with MC integration with 60 samples takes over 23 times as long. As the performance of the GMM-UB method is almost as good as the GMM-MC method, it should be preferred.

4 Conclusions

In this paper, we presented and evaluated methods for image retrieval with features derived from JPEG2000 wavelet subband data. In particular, color and texture features were computed, which represent characteristics of the image well. We illustrated that for certain images, the modelling of texture features by GGDs has weaknesses and proposed to model them by GMMs. As shown, the probabilistic modelling of the data in terms of distributions makes it easy and efficient to compare collections. Approximate evaluation of KLD between GMMs showed good results for problematic images, combined with fast evaluation. To further improve speed and retrieval quality, methods should be researched how to choose the most suitable method for modelling subband data automatically according to the data distribution.

Acknowledgment

This work was partially funded by the German Ministry for Education and Research (BMBF) through grant FKZ 01IRB02B. We would also like to thank Neuland Multimedia GmbH for providing their interns for the collection of the relevance data.

References

1. The Benchathlon collection. <http://www.benchathlon.net/img/done/>.
2. ISO/IEC 15444-1:2000, JPEG 2000 image coding system – part 1: Core coding system, edition 1.
3. M. D. Adams. Jasper: A software-based JPEG2000 codec implementation. <http://www.ece.uvic.ca/mdadams/jasper/>.
4. J. Bhalod, G. F. Fahmy, and S. Panchanathan. Region based indexing in the JPEG2000 framework. In *Internet Multimedia Management Systems II, SPIE*, 2001.
5. J. A. Blimes. A gentle tutorial on the EM algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical Report ICSI-TR-97-021, University of Berkeley, 1997.
6. M. S. Crouse, R. D. Nowak, and R. G. Baraniuk. Wavelet-based statistical signal processing using hidden markov models. *IEEE Transactions on Signal Processing*, 46, 1998.
7. M. N. Do. Fast Approximation of Kullback Leibler Distance for Dependence Trees and Hidden Markov Models. submitted to IEEE Signal Processing Letters.
8. M. N. Do and M. Vetterli. Rotation Invariant Texture Characterization and Retrieval Using Steerable Wavelet Domain Hidden Markov Models. In *IEEE Transactions on Multimedia*, volume 4, pages 517–527, 2002.
9. M. N. Do and M. Vetterli. Wavelet-Based Texture Retrieval Using Generalized Gaussian Density and Kullback-Leibler Distance. In *IEEE Transactions on Image Processing*, volume 11, pages 146–157, February 2002.
10. J. Goldberger, H. Greenspan, and S. Gordon. An efficient similarity measure based on approximations of KL-divergence between two Gaussian mixtures. *International Conference on Computer Vision (ICCV)*, 2003.
11. Z. Liu and Q. Huang. A new distance measure for probability distribution function of mixture type. *ICASSP-2000 Istanbul, Turkey*.
12. M. K. Mandal and C. Liu. Efficient image indexing techniques in the JPEG2000 domain. *Journal of Electronic Imaging*, 13, 2004.
13. T. Minka and R. Picard. Interactive learning using a ‘society of models’. In *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition (CVPR-1996)*, pages 447–452, 1996.
14. H. Müller, W. Müller, D. M. Squire, M.-M. Stphane, and T. Pun. Performance evaluation in content-based image retrieval: Overview and proposals. submitted to Elsevier preprint, August 2000.
15. M. J. Swain and D. H. Ballard. Color indexing. *Int. J. Comput. Vision*, 7(1):11–32, 1991.
16. Z. Xiong and T. S. Huang. Subband-based, memory-efficient JPEG2000 images indexing in compressed-domain. *Fifth IEEE Symposium on Image Analysis and Interpretation (SSIAI'02)*, 2002.