

Coarse-to-Fine Textures Retrieval in the JPEG 2000 Compressed Domain for Fast Browsing of Large Image Databases^{*}

Antonin Descampe^{1, **}, Pierre Vandergheynst²,
Christophe De Vleeschouwer^{1, **}, and Benoit Macq¹

¹ Communications and Remote Sensing Laboratory,
Université catholique de Louvain, Belgium
`descampe@tele.ucl.ac.be`

² Signal Processing Institute,
Ecole Polytechnique Fédérale de Lausanne, Switzerland

Abstract. In many applications, the amount and resolution of digital images have significantly increased over the past few years. For this reason, there is a growing interest for techniques allowing to efficiently browse and seek information inside such huge data spaces. JPEG 2000, the latest compression standard from the JPEG committee, has several interesting features to handle very large images. In this paper, these features are used in a coarse-to-fine approach to retrieve specific information in a JPEG 2000 code-stream while minimizing the computational load required by such processing. Practically, a cascade of classifiers exploits the bit-depth and resolution scalability features intrinsically present in JPEG 2000 to progressively refine the classification process. Comparison with existing techniques is made in a texture-retrieval task and shows the efficiency of such approach.

1 Introduction

In today's imaging applications, we observe a significant increase of the amount and resolution of digital images, together with a refinement of their quality (bit-depth). This huge and ever-growing amount of digital data requires on one hand an efficient and flexible representation for the image, allowing to reach instantaneously any part of it at a given resolution and quality level. On the other hand, application-specific techniques exploiting this flexibility are needed to guide the user in the data space and assist him in his task.

Beside a high compression efficiency, JPEG 2000 [1], the latest still-image compression standard from the JPEG committee, enables such flexibility. From a single code-stream and depending on the targeted application and available resources, various versions of specific image areas can easily be extracted, without having to process other parts of the code-stream.

^{*} This work has been supported by the SIMILAR Network of Excellence.

^{**} A. Descampe and C. De Vleeschouwer are funded by the Belgian NSF (FNRS).

In this paper, we are interested in techniques dedicated to information retrieval in the JPEG 2000 compressed domain. These techniques could indeed draw benefit from the JPEG 2000 scalability. The proposed approach is a coarse-to-fine retrieval process [2, 3].

Several papers have been written on the analysis of JPEG 2000 code-streams. Each of them proposes a feature that can be more or less easily extracted from the code-stream and that is used in a classification process. Two kind of features can basically be extracted: the ones from the packet headers that do not involve any partial decompression [4, 5, 6] and the ones based on the wavelet coefficients that require an entropy-decoding step [7, 4, 8]. Concerning header-based techniques, two features can basically be extracted: the maximum number of significant bit-planes in each block of the image [4] and the number of bytes used to entropically encode the block. The later is actually a measure of the entropy enclosed in the block and is used in [5] as a texture classification tool and a event detection system in video. [6] use the same entropy measure to select the most important parts of an image in cropping/scaling applications. Concerning wavelet-based techniques, [7] computes a vector of variance of each subband from the wavelet coefficients and uses it as the feature vector. The authors in [4, 8] propose to use the significance status of the wavelet coefficients at each bit-plane as the discriminating feature. The approach adopted in the present contribution is different and complementary from the papers mentioned above. Starting from the coarse classification that can easily be obtained from header-based techniques, we use the resolution and bit-depth scalability of JPEG 2000 to progressively extract more information and refine the classification with wavelet-based techniques. For a given image area, the amount of data that will need to be entropy-decoded is therefore directly related to the relevance of this area in the retrieval process. We introduce also a method for a progressive approximation of histogram that combines well with a partial JPEG 2000 decoding. To the best of our knowledge, only [8] suggests such progressive content extraction but the authors did not exploit this idea in a true coarse-to-fine approach implying a cascade of classifiers.

The remainder of the paper is organized as follows. Section 2 gives JPEG 2000 key elements. In Section 3, the way JPEG 2000 features are used in a succession of classifiers is explained. An application of these principles to texture retrieval is presented in Section 4. Finally, we present a summary and perspectives for this work in Section 5.

2 JPEG 2000 Key Elements

Fig. 1 shows the main steps of a JPEG 2000 compression. First of all, the image is split into rectangular blocks called *tiles*. They will be compressed independently from each other. An intra-component decorrelation is then performed on the tile: on each component a *discrete wavelet transform* (DWT) is carried out. Successive dyadic decompositions are applied. Each of these splits high and low frequencies in the horizontal and vertical directions into four subbands. The subband corresponding to the low frequencies in the two directions (containing most

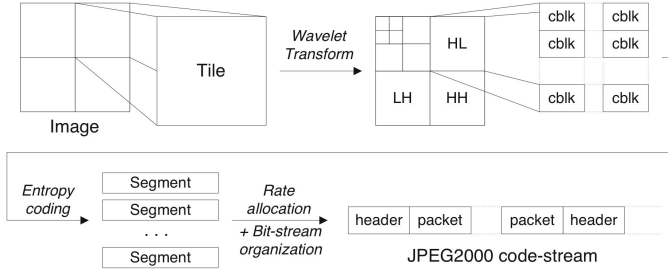


Fig. 1. JPEG 2000 algorithm overview

of the image information) is used as a starting point for the next decomposition. This DWT enables the multi-resolution feature and transforms pixels in wavelets coefficients that will be partially used in our retrieval techniques.

Every subband is then split into rectangular entities called *code-blocks*. Each code-block will be compressed independently from the others. Together with the tiling operation, this code-block creation explains the spatial flexibility available in JPEG 2000 and useful for browsing or retrieval applications. The coder used for each code-block is a *context-based entropy coder*. It removes redundancy present in the binary sequence using the probability estimates of the symbols. For a given symbol, its probability estimate depends on its neighborhood (its “context”). Practically, the coefficients in the code-block are *bit-plane* encoded, starting with the most significant bit-plane. Instead of encoding all the bits of a bit-plane in one coding pass, each bit-plane is encoded in three passes. This bit-plane-oriented encoding scheme enables the bit-depth scalability that will be used in our coarse-to-fine approach.

During the *rate allocation* and *bit-stream organization* steps, encoded code-block are scanned in order to find optimal truncation points to achieve various targeted bit-rates. Quality layers are then created using the incremental contributions from each code-block. Compressed data corresponding to the same component, resolution, spatial region and quality layer is then inserted in a packet. Packets, along with additional headers, form the final JPEG 2000 code-stream. Among other information, packet headers contain the length in bytes of each incremental code-block contribution (NB in the following), together with the number of non-zero bit-planes for each code-block (BP in the following).

3 Progressive Classification in JPEG 2000

In this Section, we present our classification scheme, which is inspired by the one proposed in [3], and the way the different scalability levels of JPEG 2000 are used. Our main objective is to locate in a large image (or a bunch of images) the areas similar to a given query. “Area” must here be understood as a spatial location in the image (i.e. represented by coordinates). A single area corresponds therefore to several JPEG 2000 packets depending on the number of resolutions

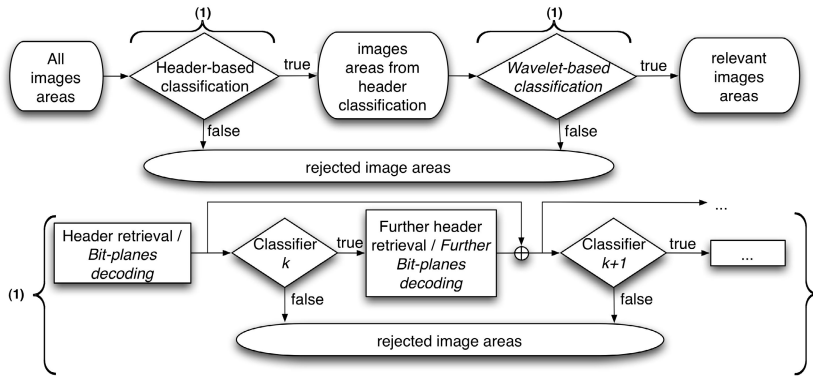


Fig. 2. The progressive classification in JPEG 2000 is made in two steps : the first one based on headers and the second one based on wavelet coefficients. Positive decisions taken by a classifier k feed classifier $k + 1$.

and quality layers specified during the encoding. Based on the hypothesis that a large majority of areas are negative (i.e., non-relevant for the query), we try to reject as many areas as possible with the smallest processing needed. To do so, several classifiers are cascaded, each of them being fed by the positive areas selected by the previous one. The main goal for each of those classifiers is to keep the false-negative rate as small as possible. The resulting high false-positive rate is not critical as deeper classifiers are designed to be more accurate. They of course require more processing but are applied on a decreasing set of areas.

As shown in Fig. 2, we defined two main steps in the process, the first one based on headers information, and the second one on wavelet coefficients. This second step requires a partial entropy-decoding while headers are directly available in the code-stream. Both steps are designed in the same way, as detailed in part (1) of Fig. 2: more information is extracted from the code-stream before each classifier, each of them building its feature-vector based on this new data and the previously extracted data. In the header-based classification, headers are progressively read, from the lowest resolution to the highest. In the wavelet-based classification, bit-planes are progressively decoded, from the most significant to the least significant, and from the lowest resolution to the highest. Wavelets coefficients are therefore better and better approximated.

4 Application to Texture Retrieval

The classification scheme described in the previous section has been applied to a texture retrieval application. We used 640 128x128 8-bit gray-scale images belonging to 40 classes, with 16 images per class. The images were obtained by splitting 40 512x512 images from the MIT Vision Texture (VisTex) database [9]. The list of texture images is the same as in [10]. All images were compressed losslessly using three wavelet transform levels and 64x64 code-blocks, like in [5].

4.1 Features Extraction

We describe here how features suited for texture characterization are derived from header-information and wavelet coefficients.

Concerning the headers, NB is a measure of the code-block entropy while BP is approximately equal to $\log_2(\|\mathbf{x}\|_\infty)$ where $\mathbf{x}$ is the vector of wavelet coefficients from a given subband. In [11], Mallat showed that these two values (source entropy and maximum modulus of wavelet coefficients) well describe the kind of singularity present in the image. As textures are mainly made of singularities from the same type, these two header-extracted values will be used just as they are in a feature-vector.

Concerning wavelet coefficients, it has been shown, like in [10], that textures are accurately modeled by the marginal densities of wavelet subband coefficients. Therefore, for a given subband, an approximation of the coefficients distribution is used as our feature-vector, this approximation being progressively refined as more bit-planes are decoded. The method used to approximate this distribution is a progressive histogram construction that suits well with a JPEG 2000 decoding scheme. While decoding a bit-plane i of a given subband, it is indeed easy to count how many coefficients among n become significant: let's define n_{i+} as the number of new positive coefficients and n_{i-} as the number of new negative ones. Then, let's define a partition of the coefficients into R disjoint intervals $\{S_1, \dots, S_R\}$ of length $\{\Delta_1, \dots, \Delta_R\}$. The probability distribution might then be approximated by the following histogram:

$$\begin{aligned} p(x) &= p_i \text{ for } x \in S_i, i = 1, \dots, R \\ p_i &\geq 0 \text{ (} i = 1, \dots, R \text{)} \\ \sum_{i=1}^R p_i \Delta_i &= 1 \end{aligned}$$

Taking R = number of significant bit-planes, $\Delta_i = 2^{i-1}$, and $p_i = \frac{n_i}{n}$, the histogram is progressively refined as each new decoded bit-plane splits the central

Table 1. Progressive histogram construction based on information collected during bit-plane decoding

Bit-plane R $2^{R-1} \leq n_{R+} < 2^R$ $-2^R < n_{R-} \leq -2^{R-1}$ $-2^{R-1} < n' = (n - n_{R+} - n_{R-}) < 2^{R-1}$
Bit-plane $R-1$ $2^{R-2} \leq n_{(R-1)+} < 2^{R-1}$ $-2^{R-1} < n_{(R-1)-} \leq -2^{R-2}$ $-2^{R-2} < n'' = (n' - n_{(R-1)+} - n_{(R-1)-}) < 2^{R-2}$
...
Bit-plane 1 $1 \leq n_{1+} < 2$ $-2 < n_{1-} \leq -1$ $-1 < n - \sum_i (n_{i+} + n_{i-}) < 1$

bin in three new and smaller non-overlapping bins. This process is illustrated in Table 1 and is referred to as *HS* in the following.

Another method to approximate the wavelet coefficients distribution has been described in [10] and consists in modeling the distribution with a *Generalized Gaussian Density* (GG). This method has the advantage to require only 2 parameters to approximate a distribution (compared to $2R$ in the histogram method). However, these $2R$ parameters are almost directly retrieved when decoding a bit-plane while the 2 parameters for the GG modeling require much more computing (a moment matching method was used for this estimation [10]).

4.2 Classification

Let sb_k and b_k be respectively the number of subbands and the number of bit-planes used in classifier k . Following feature-vectors might then be used in the classification process :

- NB_i for $i = 1, \dots, sb_k$
- BP_i for $i = 1, \dots, sb_k$
- GG_{ib_k} for $i = 1, \dots, 2 * sb_k$
- HS_{ij} for $i = 1, \dots, sb_k, j = 1, \dots, 2b_k$

To measure the similarity between two different images I and J , we used for NB_i and BP_i a *Normalized Euclidean Distance* (d_{NE})

$$d_{NE}(I, J) = \sum_{i=1}^{nsb_k} \frac{(x_i^I - x_i^J)^2}{\sigma_i^2}$$

where σ_i^2 is the variance of x_i among all images. For GG_{ib_k} and HS_{ij} , we used a sum of *Kullback-Leibler distances* (d_{KL}) between corresponding subband distributions. The K-L Distance is also known as the *relative entropy* and measures the similarity between two probability distributions.

$$d_{KL}(I, J) = \sum_{i=1}^{nsb_k} \sum_j P_i^I(j) \log_2 \left(\frac{P_i^I(j)}{P_i^J(j)} \right)$$

Based on the distance metrics defined above, five retrieval systems have been compared: the first four ones are each based on one of the four feature-vectors described above. For each of them, all the subbands from all resolutions have been used. The fifth one is a cascade of classifiers detailed in Fig. 3. A succession of 7 classifiers has been used, the first four ones based on headers, the last three ones on wavelets coefficients. NB_i and HS_{ij} feature-vectors have been preferred to respectively BP_i and GG_{ib_k} feature-vectors because they proved to give better results when used in single classifier systems. In the wavelet-based classification, only the bit-depth scalability has been used (respectively 4, 6 and 8 bit-planes at all resolutions were used in the three classifiers).

For each image, there are 15 similar items in the set of 640 images used. To compare the different schemes, the average number of relevant items among

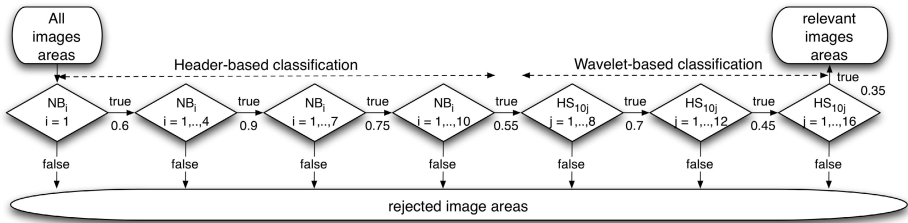


Fig. 3. Cascade of classifiers used in a texture retrieval application. Percentages of items kept at each step are indicated. False-negative rate practically achieved for each classifier is respectively, from left to right, 0.6%, 0.7%, 0.2%, 1.7%, 2.2%, 4.5% and 12.1%.

the 15 closest items has been computed for each scheme. For the cascade of classifiers, rejection rates for each classifier have been empirically chosen such that the last classifier keeps 15 items. Retrieval rates are compared in Table 2, together with the processing load required by each system. As it can be seen, the coarse-to-fine approach gives the best retrieval rate (similar to the histogram-based classifier) while only 56% of the headers have to be read and 17% of the bits have to be entropy-decoded. We also observe that despite a much heavier processing (entropy-decoding is required), a wavelet-based classification does not outperforms significantly a simple header-based scheme. This is because textures are mainly characterized by *global* parameters, such as the ones found in headers. Wavelet-based methods are expected to be much more efficient in retrieval tasks involving *local* parameters and their relative position in the considered area.

Table 2. Comparison of five classification schemes. The coarse-to-fine scheme gives results similar to the best single-classifier scheme while requiring much less processing.

	Retrieval rate (%)	Processed headers (%)	Entropy-decoded bits (%)
BP_i	35.66	100	0
NB_i	70.46	100	0
GG_i	72.97	100	100
HS_i	79.35	100	100
Coarse-to-fine	79.42	56.25	16.84

5 Conclusion and Future Work

In this paper, we have presented a new method for information retrieval inside large JPEG 2000-compressed images. This method draws benefit from the various scalability levels inherently present in JPEG 2000 code-streams and combines several retrieval techniques in a unified coarse-to-fine approach. Starting from a light, header-based classification, results are progressively refined using heavier and more efficient classifiers based on wavelet coefficients. This method has been validated in a texture retrieval application. Results showed that the proposed

approach gives similar results to the best (and heaviest) classifier considered while requiring much less processing (56% of the headers have to be processed and 17% of the bits have to be entropy-decoded).

This scalable method could be used in a large variety of retrieval tasks. Further work will investigate for which applications the proposed approach is best suited. Another perspective is the combination of this retrieval system with an adapted remote browsing strategy [12]: while navigating inside a large image, the user would be guided by the system towards relevant areas according to a submitted query. This would help achieve searching tasks faster and save bandwidth.

References

1. Boliek, M., Christopoulos, C., Majani, E.: JPEG 2000 image core coding system (Part 1). Technical report, ISO/IEC JTC1/SC29 WG1 (2001)
2. Fleuret, F., Geman, D.: Coarse-to-fine face detection. *International Journal of Computer Vision* **41** (2001) 85
3. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: *IEEE Conference on CVPR*. (2001) 609–615
4. Mandal, M.K., Liu, C.: Efficient image indexing techniques in the JPEG2000 domain. *Journal of Electronic Imaging* **13** (2004) 179–187
5. Tabesh, A., Bilgin, A., Krishnan, K., Marcellin, M.W.: Jpeg2000 and motion jpeg2000 content analysis using codestream length information. In: *Proceedings of the Data Compression Conference (DCC'05)*. (2005)
6. Neelamani, R., Berkner, K.: Adaptive representation of jpeg2000 images using header-based processing. In: *IEEE International Conference on Image Processing (ICIP)*. Volume 1. (2002) I–381– I–384
7. Xiong, Z., Huang, T.S.: Subband-based, memory-efficient jpeg2000 images indexing in compressed-domain. In 290-294, ed.: *SSIAL*. (2002)
8. Jiang, J., Guo, B., Li, P.: Extracting shape features in jpeg-2000 compressed images. In: *ADVIS '02: Proceedings of the Second International Conference on Advances in Information Systems*, London, UK, Springer-Verlag (2002) 123–132
9. MIT Vision and Modeling group: Vision texture (vistex) database (2002)
10. Do, M.N., Vetterli, M.: Wavelet-based texture retrieval using generalized Gaussian density and Kullback-Leibler distance. *IEEE Trans. Image Process.* **11** (2002)
11. Mallat, S.: *A Wavelet Tour of Signal Processing*. Academic Press (1998) 577p.
12. Descampe, A., DeVleeschouwer, C., Iregui, M., Macq, B., Marques, F.: Pre-fetching strategies for remote and interactive browsing of JPEG2000 images. to appear in *International Conference on Image Processing*, 2006. *ICIP '06*. (2006)