Université d'Ottawa • University of Ottawa

CONDITIONAL ENTROPY-CONSTRAINED VECTOR QUANTIZATION OF CHROMATIC INFORMATION IN DOCUMENT IMAGES

Vincent Guillopé

A Thesis submitted to the Faculty of Graduate Studies and Postdoctoral Studies in partial fulfillment of the requirements for the degree of Master of Applied Science, Electrical Engineering

April 2002

Ottawa-Carleton Institute for Electrical and Computer Engineering
School of Information Technology and Engineering
University of Ottawa
Ottawa, Ontario, Canada

To my family.

Contents

List of Figures

Abstract

We present in this thesis original research on Conditional Entropy-Constrained Vector Quantization (CECVQ) of chrominance information in document images. First, we review the existing image compression formats that could be used for document images: JPEG 2000, DjVu, LDF and PNG. Then, we develop the CECVQ algorithm to adapt it to image quantization and we show its excellent distortion–rate performance. Finally, we develop our own coding scheme targeted at the compression of chrominance information and based on the piecewise-constant model. Color space transformation, CECVQ and context-based entropy coding form the skeleton of our scheme. We also show the strengths and the weaknesses of this new coding scheme, and suggest possible improvements and other approaches.

Acknowledgements

In first place, I would like to thank my supervisor Dr. Éric Dubois for his guidance and his time. Working under his supervision brought me a big glimpse of what is an enthralling scientific research.

Also I must thank The National Archives of Canada for their gracious help in providing original archive document images.

Finally, I thank you, my dear parents, for allowing me to spend this rich and unique research time in Canada, supporting me however far I could be.

Chapter 1

Introduction

1.1 Problem understanding

Public Archives would like to give access to electronic versions of their huge collection of documents to researchers and the general public. The storage of such an immense electronic library requires appropriate image coding of the electronic versions of the archive documents to be able to minimize the size requirements. The smaller a document image is, the less storage space it requires and the faster it can be downloaded by an Internet user. Thus, by decreasing the size of a document, we make a document cheaper to store and faster to access, two key preoccupations for on-line libraries.

The image properties such as colors, spatial resolution, image composition are of prime importance in the choice or the design of an appropriate document coding scheme. The typical document we are interested in, *i.e.* the archive document, has a quite homogenous background with usually smooth intensity variation and few colors. The foreground is constituted of a hand-written text that can have different colors. Other components can be present such as a wax seal or a library stamp.

1.2 Motivation of the study

There exist numerous different image formats that we could use to encode our document images. Each of them has different characteristics with regard to average achievable size or visual deterioration. For our study on how to encode archive document images, we studied the latest image and document coding formats in order to determine their characteristics in the particular case of archive document coding. The most recent and efficient image coding formats are JPEG 2000 for natural image coding and PNG for palette-based image coding. Recent document image coding formats are DjVu by LizardTech and LDF by LuraTech. In our literature review, DjVu seems to be the most promising format for document image coding. However, we present the result of our own compression tests on the formats cited above to determine the most adapted format for our specific needs.

However, these existing formats fail to provide both high compression and reasonable visual quality for researchers. Therefore, we propose a new method based on the typical characteristics of such archive documents. The overall simplicity of these documents should enable us to define a more efficient method by taking advantage of their specific characteristics. The main idea is to quantize the vectorial chrominance information into an optimal set of values and to entropy encode it, while the intensity information of the image can be encoded with an existing method such as JPEG 2000.

1.3 Thesis development

In Chapter 2, we first review and point out the strengths and the weaknesses of the latest available formats likely to be used in the compression of archive document images: JPEG 2000, DjVu, LDF and PNG. Then, in Chapter 3, we address the general problem of codebook design and, more specifically, we adapt the Conditional Entropy-Constrained Vector Quantization (CECVQ) algorithm recently introduced for speech and one-dimensional synthetic sources of data to the two-dimensional chrominance information. Finally, we develop in Chapter 4 a new compression method based on the

original idea of vector quantization and entropy coding of the vectorial chrominance information and also address the problem of color space transform and context-based arithmetic coding. Current results, future expectations and guidance for future research are suggested from the numerous experiments we ran and from the experience developed in this area.

Chapter 2

Using Recent Document Coding Methods

Before we propose our own method, we review recent image compression schemes that could be used for the compression of archive document images.

2.1 Document image characteristics

Before discussing available methods for image compression, we must analyze the image characteristics to be able to determine which features of a method are suitable and which ones are not.

In our case, we are interested in scans of archive documents. We call these scans document images. These images would eventually be available on Internet for consultation by remote researchers who then would not have to travel to Ottawa (or wherever the document is held) to study a specific document. Thus, these images must be as similar as possible to the original paper document, in order to keep their value for the researchers. We would like to have a close reproduction of the writing, of the colors of the paper and of the ink, and keep all these little details which give

a *cachet* to a document. But we also want an encoding scheme that makes these documents much smaller than their uncompressed size.

A typical archive document is scanned at 300dpi. Typed text and handwriting have sharp contour and abrupt transitions with the background, which correspond to high spatial frequencies and need high resolution to be closely captured. The resulting image is typically a 2500x4000 pixel color image, which needs around 30 megabytes of storage in an uncompressed format. As a library can own many thousands of these archive documents, we can see the urge in finding an appropriate coding scheme to greatly reduce the storage requirements.

Regarding their visual characteristics, the archive documents we are considering are fairly simple. The nature of the paper gives a background with a rather homogenous color and texture. The writing has similar properties. Some other details can also be present such as a library ink seal or a wax seal. The overall number of color is limited to three or four: the background, the writing ink and some other details. We can then theorize that only a few colors could be used to represent the entire image.

We will now review the most recent image formats adapted to the encoding of such image documents. We will study the JPEG 2000 format, also known as the successor of the popular JPEG. The DjVu and the LDF are two image formats related to JPEG 2000 that particularly aim at the encoding of image documents. Finally, the PNG format can encode images in a palette-based format with a reduced set of colors.

2.2 Available recent schemes

2.2.1 JPEG 2000

JPEG 2000 represents the latest advances in image compression technology and aims to bring the best techniques for the compression of "different types of still images (bilevel, gray level, color, multicomponent) with different characteristics (natural images, scientific, medical, remote sensing, text, rendered graphics, etc.) allowing different imaging models (client/server, real-time transmission, image library archival.

limited buffer and bandwidth resources, etc.)" [19]. This citation sums up very well the broad ambition of JPEG 2000.

The JPEG 2000 Part 1 standard, which defines the core functionalities, was released in late 2000. Analysis of its functionalities and performance are spreading little by little in the literature. A complete introduction and performance evaluation of JPEG 2000 can be found in [15, 19, 11].

The compression technology of JPEG 2000 is based on the Discrete Wavelet Transform (DWT), followed by scalar quantization, context modelling, arithmetic coding and post-compression rate allocation. The particular organization of the coded data in codeblocks, layers and packets enables the JPEG 2000 format to have a rich set of features.

First of all, JPEG 2000 supports component transforms, such as color space transforms. Transforms from RGB to YCbCr and to an approximation of the YUV space are defined in Part 1 of the standard. By decorrelating the different components, color space transforms allow more efficient quantization of the wavelet coefficients. In the case of few colors, the color information is concentrated in the (Cb, Cr) or (U,V) components, allowing high compression of these subbands while the luminance part can be allocated more rate.

JPEG 2000 also supports tiling. This allows compression of very large images using hardware with limited capability. It also gives the possibility to manage a Region Of Interest (ROI) that can be encoded at a different compression ratio than the background.

Finally, multiresolution and scalability features are provided by the inherent organization of the coded data. This offers very practical way of handling images especially across networks. A degraded version of the original image can be reconstructed with any quantity of received data. If the connection is lost during the data transfer, the end user will still be able to see a degraded version of the transmitted image. The more bits that are received, the better the quality. Another application could be the multicasting of a unique image to different users. Each user can receive the image

with a different quality according to their network bandwidth or privilege rights.

All these features make the JPEG 2000 format very likely to spread quickly. In our particular case of encoding document images to make them available across a network, these features can be a great advantage to manage the storage and the distribution of a huge quantity of image data.

2.2.2 Mixed Raster Content formats

The Mixed Raster Content (MRC) formats were developed for the encoding of those images whose type lies between the natural image and the graphic image. They aim particularly at newspaper and magazine scanning and more generally compound documents. The natural properties of these document images lie in the background and in the picture that are illustrating the articles. The graphical properties lie in the text and the schema.

The decoder scheme under the designation of Mixed Raster Content document coding was standardized in [20] and an analysis of this type of documents was accurately done by De Queiroz *et al.* in [7, 8]. The basic idea is to divide the document image in different areas according to their properties. Each area is isolated in a layer. Each layer is encoded with an adapted method.

A common decomposition is the three layer MRC document, which contains a background layer, a foreground layer, and a binary mask layer. At each pixel the value of the binary mask is used to select between the foreground and the background pixel. Typically, the foreground and the background layer could be encoded at 100 dpi with a JPEG 2000 encoder or some other wavelet based encoders, while the binary mask would be encoded at 300 dpi with the state-of-the-art in terms of binary compression such as the upcoming JBIG2. The background would hold color and natural scene information. The foreground would hold color information of text and graphics. And the mask would efficiently hold spatial high-frequency information involved in the edges and contours of text and graphics.

MRC coding is envisaged to be included in JPEG 2000 Part 6, which will be

released in the near future. But MRC coding schemes already exist in the DjVu format [4] developed by the AT&T Labs (DjVu was then bought by the LizardTech company) and in LDF (LuraDocument Format) [2] developed by the German company LuraTech (LuraTech is also an important contributor to the JPEG 2000 format, especially through its own wavelet coder LuraWave).

They both use the same multi-layer approach described previously, but different coders. The 1-bit coder of DjVu is JB2. AT&T's proposal for JBIG2 and performs better than the 1-bit coder of LDF. On the other hand, the wavelet coder of LDF is the LWF coder (LuraWave Format), which is the LuraTech implementation of JPEG 2000, and performs better than the IW44 wavelet coder of DjVu. The true difference between the two formats lies on their segmentation policy. They have "segmenters" with different performance. Segmenters perform high-contrast analysis on a color image and split it into foreground and background layers. It would seem that DjVu has a more advanced text segmenter than LuraDocument.

DjVu and LDF are available as plug-ins and stand-alone applications for demonstration to commercial purposes.

2.2.3 PNG

Portable Network Graphics (PNG) [21] is a W3C recommendation for coding of still images, which has been elaborated as a patent-free replacement for GIF, while incorporating more features. It is based on a predictive scheme and entropy coding. The prediction is done on the three nearest causal neighbors and there are five predictors that can be selected on a line-by-line basis. The entropy coding uses the Deflate algorithm of the popular Zip file compression utility, which is based on LZ77 coupled with Huffman coding. PNG is capable of lossless compression only and supports gray scale, palette-based color and true color, an optional alpha plane, interlacing and other features.

As for GIF, it is a lossless compression scheme. It can be feared that only bad compression ratio can be achieved, but the efficiency of this format first takes place in

the use of filters and pre-processing. A very simple example of filter is color reduction through palette manipulation.

PNG was developed for use with graphic images. Since we can consider that our type of image lies somewhere between the natural image and graphical image, PNG is a format that is worth studying.

2.3 Performance

For our tests, we use the following software:

JPEG 2000: JasPer 1.500.3 developed in a collaborative effort between the University of British Columbia and Image Power, Inc.. Free open-source software written in C and JPEG 2000 reference software [1].

DjVu: Solo 3.1 from LizardTech, Inc., available as a demonstration software.

LDF: IrfanView 3.61 with the LDF plug-in, written with the LDF SDK of LuraTech by I. Skiljan (Vienna University of Technology).

PNG: The Gimp 1.2.2. Free open-source image processing software, known as having one of the best implementations of the PNG format.

2.3.1 Distortion–rate performance

Throughout this thesis, we call distortion the root mean square error between the encoded image and the original image. In this section, the rate is expressed in output bits per input bit. This convention was chosen because of the "rate" argument of the JasPer program, which is indeed the inverse of the compression ratio. A rate of 0.1 will therefore indicate a compression ratio of 10:1.

We give in Fig. 2.1 the distortion–rate performance of JPEG 2000 on the two test images we will use throughout this document and shown in Appendix A. The

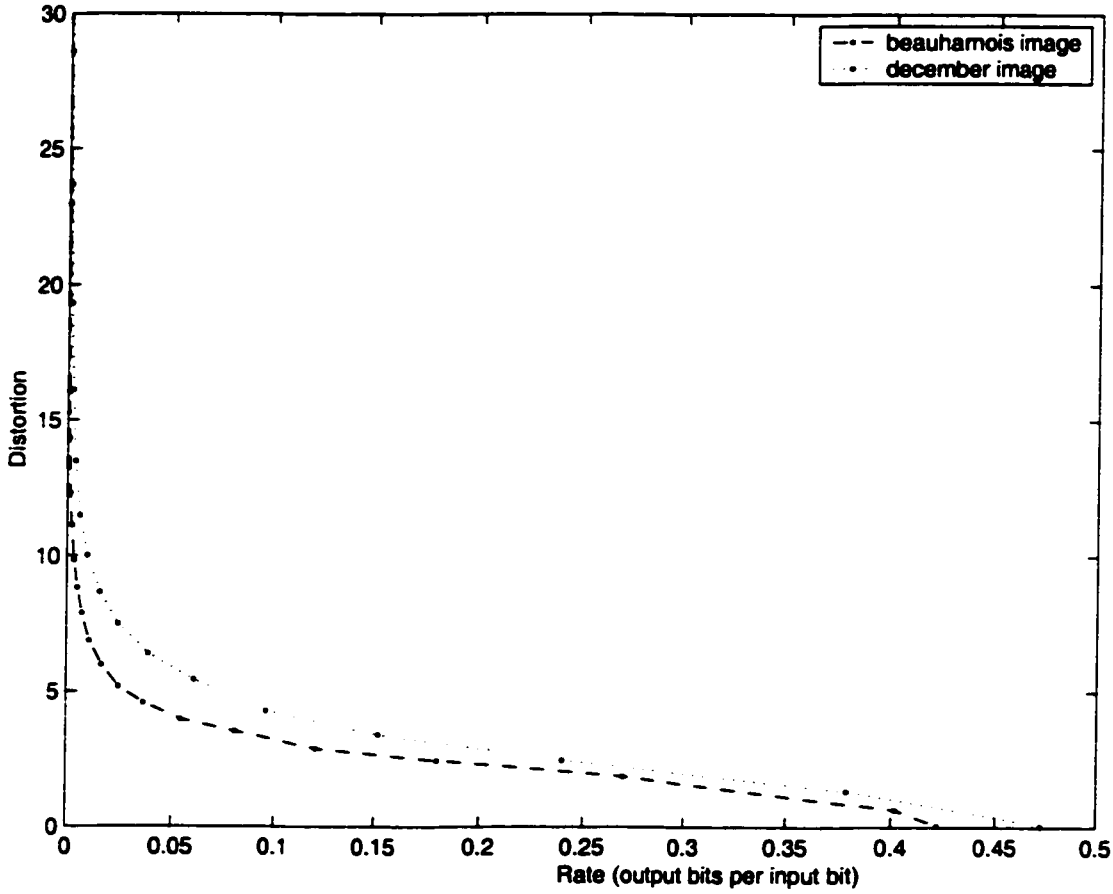

Figure 2.1: Distortion–rate performance of JPEG 2000 on the beauharnois and the december images

shape of the curve shows that the coder has good performance until very low rates. Distortion starts increasing considerably only below rate 0.02.

We could not draw the distortion–rate curves for DjVu and LDF. The coders provided on a free basis by their respective companies do not allow to specify a target rate. Instead, a type of document and eventually (LDF) a quality level (good, medium, low) are the only options offered.

For PNG tests, we encoded the "beauharnois" image in a palette-based format with a decreasing number of index entries. Even if PNG is designed as a lossless format, it can be associated with filters and result in a lossy compression scheme. A basic filter is color reduction from 16 million to 256 down to 3 or 2. We obtain the distortion–rate curve and we compare it to JPEG 2000 in Fig. 2.2. It is clearly shown that the PNG format used with color reduction does not perform as well in the distortion–rate sense.

2.3.2 Compression ratio and visual quality performance

For lossy compression of color images, JPEG 2000 is the cornerstone. The basic JPEG 2000 standard Part 1 provides excellent visual performance for documents with compression ratios as high as 100:1. In this range, the visual difference is undetectable or small and similar to a blur effect. Published studies [15, 19, 11] emphasize the quality improvement at high compression ratios of JPEG 2000 over other existing image encoding schemes. However for compression ratio higher than 100:1, the quality quickly drops, especially around areas with high spatial frequencies, such as text. The blur is too great to read any normal sized text. In general, sharp edges start being surrounded with artifacts due to the cut of high frequencies (Gibb's phenomenon).

The use of the MRC formats DjVu and LDF enable us to achieve even larger compression ratios while preserving the text readability. The background is strongly blurred as it is for very high compression ratios using wavelet decomposition. The color of the text is also almost completely homogenous and the text contour is more or less well detected.

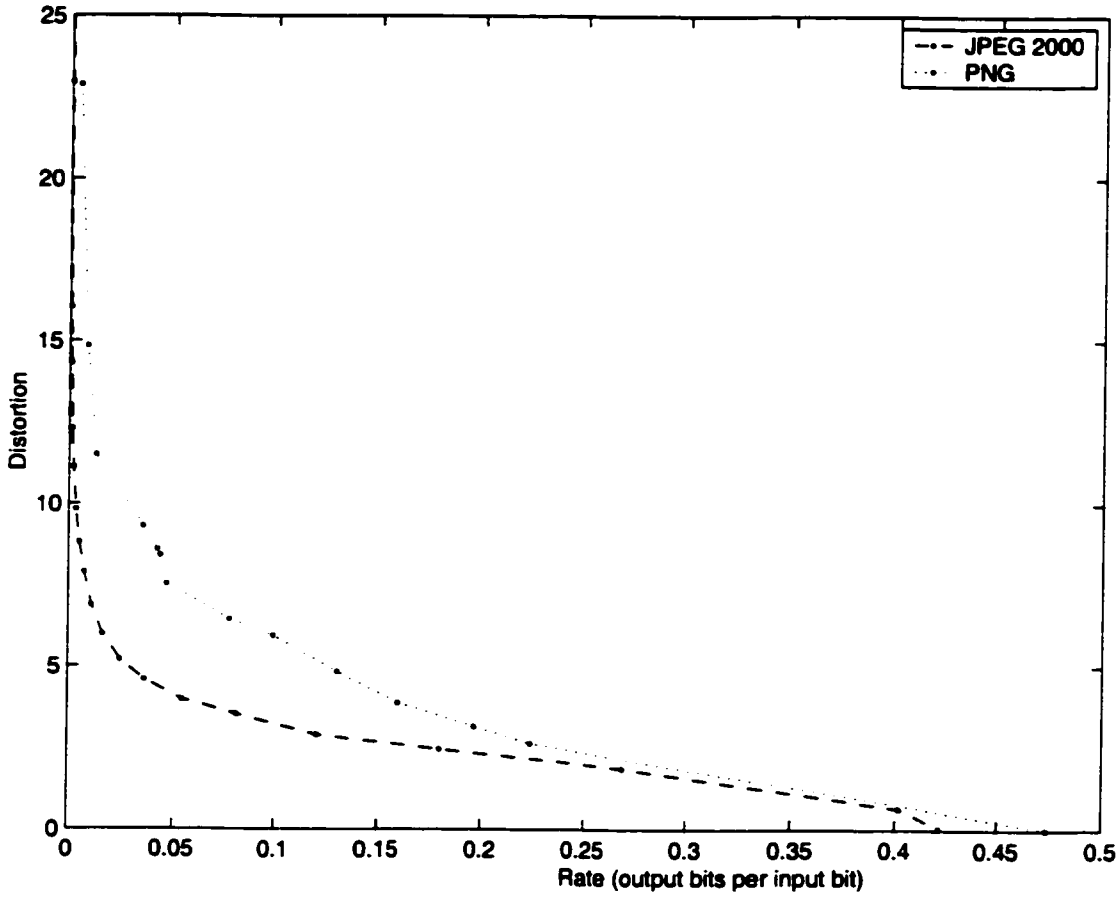

Figure 2.2: Distortion–rate performance of PNG compared to JPEG 2000 on the beauharnois image

The DjVu format reaches compression ratios up to twice as large as LDF achieves, but this seems to be mainly a question of configuration. The achieved compression ratio depends on the compression of the background, on the quantity of information in the foreground – that is how much text was detected and put there – and on the homogeneity of the mask.

At a similar visual quality, the analysis is more complex. For a quality that we could eventually use to encode our image documents, we require a good text detection, so that it can have good legibility, and we would like to keep a good quality for the representation of particular details such as a wax seal or a library stamp. As shown in Fig. 2.3, it appears that LDF has a better text detection and can also isolate some important details such as the wax seal. But at the same time, many little and thin irrelevant details are put in the foreground and increase the complexity of the mask, leading to a lower compression ratio. In the case of DjVu, not all the handwriting is successfully detected and nothing of the wax seal. However, the writing appears thicker and is then a little easier to read. The mask is also more homogenous.

Knoll, from the National Library of the Czech Republic, presented in [12] some tests performed on the JPEG 2000, DjVu and LDF formats for the purpose of image document encoding. According to his analysis, the best lossless compression format for color images is PNG and JPEG 2000 is also expected to yield good performance in the lossless domain. For lossy compression of color images, JPEG 2000 is the state-of-the-art of the image standards. As for their comparison between DjVu and LDF, he concludes that DjVu has the best binary encoder and LDF the best wavelet encoder. In his tests, DjVu performs up to twice as good. DjVu stores less information in the foreground than LDF. It then has a simpler mask that can be encoded at lower rate. The main difference between the two MRC formats are their segmentation policy. Knoll seems to give some preferences to the DjVu format because it achieves higher compression ratios.

The DjVu format seems then to have a slight advantage on LDF. Exploiting the strengths of both formats could lead to a better MRC format. But this is also the

subject of JPEG 2000 Part 6, for which DjVu and LDF seem to be research work.

As for PNG, the visual characteristics of the image obtained by reduction of the color palette are interesting and full of lessons. Figure 2.4 shows the results of color reduction on a portion of the "beauharnois" image. With 16 color entries in the palette we have a very good reproduction of the original 16 million color image. PNG compression of this image gives a ratio of 10:1. For 4 colors, some details starts missing, especially in the wax seal, but the text is still as legible as the original image. The background has a homogenous color, thus the paper texture is lost. The compression ratio is 80:1. These results are interesting because they actually show that the type of images we are using only have a few colors indeed. However the compression ratio is quite poor. Color reduction followed by PNG encoding is then not a very good candidate for image document coding. But it does give a good idea on how to design a new scheme different from the MRC formats.

2.3.3 Limitations and possible improvements

As for JPEG 2000 Part 1, the limitations are obvious. The compression of text and in general of objects with sharp edges such as handwriting produces large artifacts surrounding these edges and lowers the reading quality. All of this is due to the fact that JPEG 2000 suppresses or modifies some important high-frequencies in its compression and quantization scheme.

Possible improvements include different compression ratios in different areas of a compound document. Text and drawing would be compressed with low compression ratios while higher compression ratios can be achieved with the background and the picture. Efficient segmentation is then required. But this is the purpose of JPEG 2000 Part 6, which intends to address this problem thoroughly. However, Part 6 is still under standardization at the time of writing.

Limitations in DjVu and LDF include bad detection and non detection: bad detection if text was detected where there is no text and non detection if text was not detected where there is text. While the former can have an influence of the quality of

Figure 2.3: Comparison between DjVu (330:1) and LDF (170:1) for an excerpt of the beauharnois image (from left to right: original, DjVu and LDF)

Figure 2.4: PNG compression tests (From left to right: original, 16 colors (10:1), 4 colors (80:1))

the illustration by breaking it through different layers and different coding schemes. the latter will usually make the text illegible, because it is encoded at lower resolution and with a wavelet decomposition instead of being encoded in the mask.

These two formats work well with newspapers in which the text is more easily predictable and detectable. But with handwriting, the results are not as good, and the segmentation fails to detect correctly the handwriting.

Another point is that the companies which are developing those formats are so eager to prove to the world that they can achieve very large compression ratios, that they actually force a strong compression for the foreground and the background in their demonstration software. The required quality for our document was then not achievable. The color of the handwriting ink is made completely uniform and the texture of the paper is lost. These details might be valuable information for researchers!

Chapter 3

Conditional Entropy-Constrained Vector Quantization of Images

Quantization is a key matter when it comes to defining a lossy compression scheme. Here, we will study and adapt to image quantization a new quantization algorithm based on conditional entropy-constrained vector quantization, first developed for speech data.

3.1 Algorithm basics

3.1.1 Theoretical context

Information Theory

The Conditional Entropy-Constrained Vector Quantization (CECVQ) algorithm was first presented by Chou and Lookabaugh in [5] as a new method for the compression of linear predictive coefficients of speech. It is based on the Entropy-Constrained Vector Quantization (ECVQ) algorithm developed by the same authors in [6]. Both

repose on the same idea: to quantize data with the lowest distortion achievable for a target entropy rate of the quantized data.

The distortion–rate curve is supposed to range from null rate with a finite distortion (the entire data is quantized in one codeword) to a null distortion with a finite rate (the rate of a codebook that holds one codeword for each different vector in the input data). From one end to the other, the size of the codebook naturally varies from one to a finite number (the number of different vectors in the input data).

According to information theory [3], coders exist that can encode data to a rate arbitrarily close to its entropy rate. Two well-known coders which have this property are the Huffman coder and the arithmetic coder. They are called entropy coders. Thus, by choosing the entropy rate at which we quantize our data, we actually choose the rate to which we want to encode our data, supposing a perfect coder. What we call here rate is expressed in bits per symbol. A symbol is for instance a codeword or a label. A rate of 0.1 means that 0.1 bit is required to encode one symbol.

Now, the difference between ECVQ and CECVQ is just the considered entropy. In ECVQ, the quantization is constrained by a zeroth-order entropy, that is, by the occurrence probabilities of the codewords – a codeword represents a quantization level and a codebook is a set of codewords. In CECVQ, the quantization is constrained by a higher-order entropy. Without entering deeper into information theory, we just recall that higher-order entropy calculation takes into account the eventual correlation between the input data vectors. If the input data is not correlated, zeroth-order and higher-order entropies are the same and the CECVQ algorithm reduces to the ECVQ algorithm. However, in the case where the data is highly correlated, such as in images, the entropy rate significantly decreases through higher orders. Then an entropy coder that takes advantage of it by matching high-order entropy rate will be able to compress substantially better than an entropy coder matching zeroth-order entropy. Such an entropy coder is called a context-based entropy coder, since high-order entropy is computed by taking into account a context, also called a neighborhood in image processing.

Lagrange Multiplier Theory

At the origin of the ECVQ and CECVQ algorithms is the Generalized Lloyd Algorithm (GLA) developed in [14]. The GLA performs a vector quantization that minimizes the distortion between the input data and the quantized data. It is a special case of the ECVQ and CECVQ algorithms for a null constraint on the entropy. All of these algorithms are iterative descent algorithms. An objective function is made to decrease at each iteration until a certain threshold decided by the user is reached. For the GLA, the objective function is the total distortion between the input data and the quantized data. For the ECVQ and the CECVQ algorithms, the objective function J is a balanced sum of the total distortion D and the total entropy H: $J = D + \lambda H$, where λ is called the Lagrange multiplier. Under Lagrange multiplier theory, the minimization of this J function with a fixed λ is equivalent to a minimization of the distortion D with the constraint $H \leq H_{target}$, where H_{target} depends on the value of λ. This is exactly what we want to do with the ECVQ and the CECVQ algorithms.

3.1.2 Generalized Lloyd Algorithm

As the GLA is a simplified version of the ECVQ and CECVQ algorithms, we recall here the main lines of the algorithm.

The GLA takes two main parameters to quantize an input data set: an initial codebook and a threshold value. The size of the initial codebook is equal to the number of quantization levels since each codeword is a quantization level. Then, the bigger the initial codebook, the less distortion we will have. The threshold value is a value that determines when the iteration process must be stopped. Usually, the algorithm is stopped when the decrease rate – for instance, the difference between the distortion values of the previous and the actual iterations divided by the present distortion value – drops under the threshold value.

The algorithm itself can be decomposed into three parts:

Step 1: Each input vector is mapped to the nearest codeword using the distortion distance.

In the case of image processing, the image can be scanned in raster order but not necessarily. The distortion distance is usually the Euclidian distance, but other distances might be used with slightly different results.

Step 2: Each codeword is moved to the centroid (or statistical mean) of all input vectors that were assigned to it.

Step 3: The overall distortion is recomputed. The algorithm is stopped if the distortion decrease rate falls below the chosen threshold value.

The main strength of the GLA is its overall simplicity. It is very easy to implement and to control. However, it also has several drawbacks. Because it uses a greedy approach, the solution is suboptimal and moreover is very sensitive to the initial conditions, that is, the initial codeword values. This raises the question of how to choose these initial values. The convergence is also not very fast and requires several passes on the entire data set to achieve a fair convergence. This proves to be especially time-consuming and unpractical for large quantities of data such as the images we are considering.

3.1.3 Recursive Splitting Algorithm

As we just pointed out, we need to choose the initial codeword values. This is also true for the ECVQ and CECVQ algorithms, but as we will see later on, these algorithms usually use a codebook built by the GLA (the ECVQ and the CECVQ algorithms reduce to the GLA for $\lambda = 0$). Then, the choice of the initial codebook is important.

Many techniques exist based on random choices of the initial codewords or histogram analysis. For our GLA implementation, we used a Recursive Splitting Algorithm (RSA) and we will refer to this implementation as the RSA+GLA implementation. The principle is fairly simple and efficient:

Step 0: We start with a codebook of size one. The unique initial codeword value is chosen as the mean of the data set.

Step 1: We add a small random perturbation to the codeword of the cluster that has the most distortion, thus increasing the codebook size by one.

Step 2: The GLA is run on the new codebook, leading to a new partition of the input data set and an improved codebook. The distortion of each cluster of the partition is computed.

Steps 1 to 2 are repeated until the appropriate codebook size is reached. This algorithm can be refined by using parallel codebooks. Each codebook is obtained with a different perturbation and the GLA is performed on each one. The codebook that has the least distortion is chosen for the next step.

Although the RSA+GLA is very easy to implement and gives very good results, it is very slow. As it was intended to be the very first step of the ECVQ and CECVQ algorithms, we decided to look for another segmentation algorithm that would give an acceptable codebook in a reasonable amount of time.

3.1.4 Binary Tree Partitioning

We consider an alternative to the GLA that is less efficient in terms of distortion minimization but offers several advantages : the Binary Tree Partitioning (BTP). This algorithm can be used to create a reduced color palette of an image by segmenting the RGB space in a pre-determined number of clusters, as described in [16]. We adapted the algorithm described in [16] to our needs and we apply it to the segmentation of input data composed of 2-dimensional vectors (chrominance vectors of the input image, as we will see later in Section 4.3).

The algorithm is briefly presented here. A full explanation of the algorithm can be found in [16]. The BTP algorithm takes as sole parameter the desired number of codewords. The segmentation follows a binary tree structure, whose root represents

the whole data set. In the tree, each node represents a part of the data set and is called a cluster. Two children of a given node are a binary partition of their parent node. Therefore, the leaf node set is a partition of the whole data set. The centroids of the partition clusters are the codewords of the designed codebook.

Step 0: The root node of the tree is associated with the whole input data set.

Step 1: The mean and the covariance matrix of each new leaf are computed.

Step 2: Using eigenvector decomposition of the covariance matrix, the direction in which the cluster variation is greatest is determined. The corresponding eigenvalue is a measure of this variation.

Step 3: Among all leaf nodes, we split the one which has the highest variation (the largest eigenvalue) with a plane which is perpendicular to the eigenvector direction and passes through the cluster mean (the centroid or codeword value).

Step 4: 2 new leaves are then obtained. The mean of the clusters are the two new codewords, which replace the one of the split leaf. The codebook size is then increased by one.

Steps 1 to 4 are repeated until the desired codebook size is reached.

By means of this algorithm, the input data is segmented with a logarithmic complexity, faster than the linear complexity of the GLA. On the other hand, the distortion obtained with the BTP algorithm is usually a little higher than with the GLA+RSA. GLA can also be run on top of BTP (we will call this implementation BTP+GLA). Even then, RSA+GLA performs slightly better than BTP+GLA.

Applied to image segmentation, we found that the structuring property of the BTP algorithm helps to have small but distinct areas represented by a codeword. As we will see latter in Section 4.3, each cluster will be intended to represent a color. Even if a small area is very different from the rest of the image, the GLA might not represent it with a codeword because it will hardly contribute to the overall distortion. But, an inappropriate quantization of such an area can result in unwanted visual artifacts.

For both its time savings to design a codebook and its more accurate description of the structure of images, we will use the BTP algorithm to design the initial codebook that will be used by the CECVQ algorithm. We choose not to use the BTP+GLA, since the very first step of the CECVQ algorithm will include GLA iterations in any case.

3.1.5 Comparison of codebook design algorithms

We report here a quantitative comparison between the RSA+GLA and the BTP algorithm for two images. The first image is a typical archive document scanned at 300dpi. Such images have typically 2500x4000 pixels (10 million pixels), and need around 30 megabytes of storage to store in an uncompressed format. The second image is the "beauharnois" image (about 300,000 pixels), which was extracted from a bigger archive document image.

The RSA+GLA and the BTP algorithm were run on a Pentium III 800 MHz with 256 MB of RAM. We quantized each image with the RSA+GLA and the BTP algorithm with a codebook size of 4 and 8. We computed the distortion introduced in each case. The distortion difference between the RSA+GLA and the BTP algorithm is calculated as the difference of the values divided by their average value. From the log file of the RSA+GLA quantization, we also determine the minimum number of iterations necessary to match the performance of the BTP algorithm and we denote it as the RSA-GLA*. However, this extrapolation is possible only when RSA+GLA outperforms the BTP algorithm, which is not always true. The results are summarized in Table 3.1.

We clearly see from these results that the overall performance of the BTP algorithm completely outperforms the RSA+GLA and the RSA+GLA*, first, because it is much faster, and second, because in terms of distortion it is either not far behind the RSA+GLA or it is actually better! When the RSA+GLA falls into a local minimum, the BTP algorithm may not, because it is not a greedy iterative descent algorithm.

		10 million pixels		300,000 pixels	
		4 codewords	8 codewords	4 codewords	8 codewords
Distortion	BTP	**2.01214**	1.55628	2.00110	**1.37066**
	RSA+GLA	2.05083	**1.52429**	**1.95270**	1.55654
	RSA+GLA*[1]	N.A.	1.54943	2.00611	N.A.
	Difference (%)	-1.9	2.1	2.4	-12.7
	Difference*[1] (%)	N.A.[2]	0.44	-0.25	N.A.[2]
Time (s)	BTP	32	70	0.40	0.71
	RSA+GLA	104	267	3.73	9.02
	RSA+GLA*[1]	> 104	199	2.92	> 9.02
	Ratio	3.3	3.8	9.3	12.7
	Ratio*[1]	> 3.3	2.8	7.3	> 12.7

[1] If the RSA+GLA outperforms BTP, GLA iterations are reduced to match BTP performance.
[2] More GLA iterations would actually be needed to match BTP performance.

Table 3.1: Comparison of the BTP algorithm and the RSA+GLA

In the RSA+GLA, a lot of time is spent in the RSA part. A GLA iteration must be run after each new codeword. Also, when the desired number of codewords has been reached, more GLA iterations are still needed before the distortion stops decreasing noticeably.

On the other hand, the BTP algorithm makes use of the statistical properties of the image to create a codebook that has good distortion properties by nature. Further more, for each new codeword, only the pixels of the cluster being split need to be considered, dramatically reducing the time of the segmentation. This explains why the BTP algorithm is 3 to 13 times faster than the RSA+GLA in our test.

3.2 Algorithm development

Until now, we have presented two algorithms to obtain an initial codebook. With the first one, RSA+GLA, we can achieve a codebook with a nearly optimal distortion at the cost of lengthy computations. With the second one, BTP, we achieve a codebook with similar distortion but in a fraction of the time of the other algorithm and we also expect to capture some of the intrinsic structures of the image providing a visual benefit in the subsequent algorithms.

3.2.1 Entropy-Constrained Vector Quantization

Before studying the CECVQ algorithm used in our coding method, we will consider the ECVQ algorithm, which is very similar and is at the origin of the CECVQ.

Algorithm Description

The ECVQ algorithm actually looks very much like the GLA, except that we want to minimize an objective function $J = D + \lambda H$ instead of the overall distortion. The ECVQ algorithm can be described as follows:

Step 1: Each input vector in the data set is mapped to the nearest codeword in the sense of the objective function. In other words, the input vector x is mapped to a codeword c_x that satisfies: $\forall c,\ j(x, c_x) \leq j(x, c)$.

Step 2: The probability model is updated by counting the number of occurrences of each codeword.

Step 3: Each codeword is moved to the centroid of the set of input vectors mapped to it.

Steps 1 to 3 are repeated for a constant value of λ until the decrease of the J function drops below a threshold defined by the user. We call the function j used above a partial objective function and we define it as $j(x, c) = d(x, c) + \lambda h(c)$, where $d(x, c)$ is the distortion between the input vector x and the codeword c and $h(c)$ is the self-information carried by c, that is $h(c) = -\log_2 P(c)$, $P(c)$ being the probability of occurrence of c. We also have $J = \sum_x j(x, c_x) = \sum_x d(x, c_x) + \lambda \sum_x h(c_x)$.

The ECVQ algorithm is performed for several values of λ. Each value of λ corresponds to a value of the entropy rate of the quantized image. However, there is no simple relation between λ and the entropy rate. Thus, to achieve a given entropy rate, the λ value must be progressively increased from zero until the target entropy rate is achieved. Eventually, some codewords of the initial codebook will be dropped to achieve a smaller codebook matching the required entropy rate. Fig. 3.1 shows

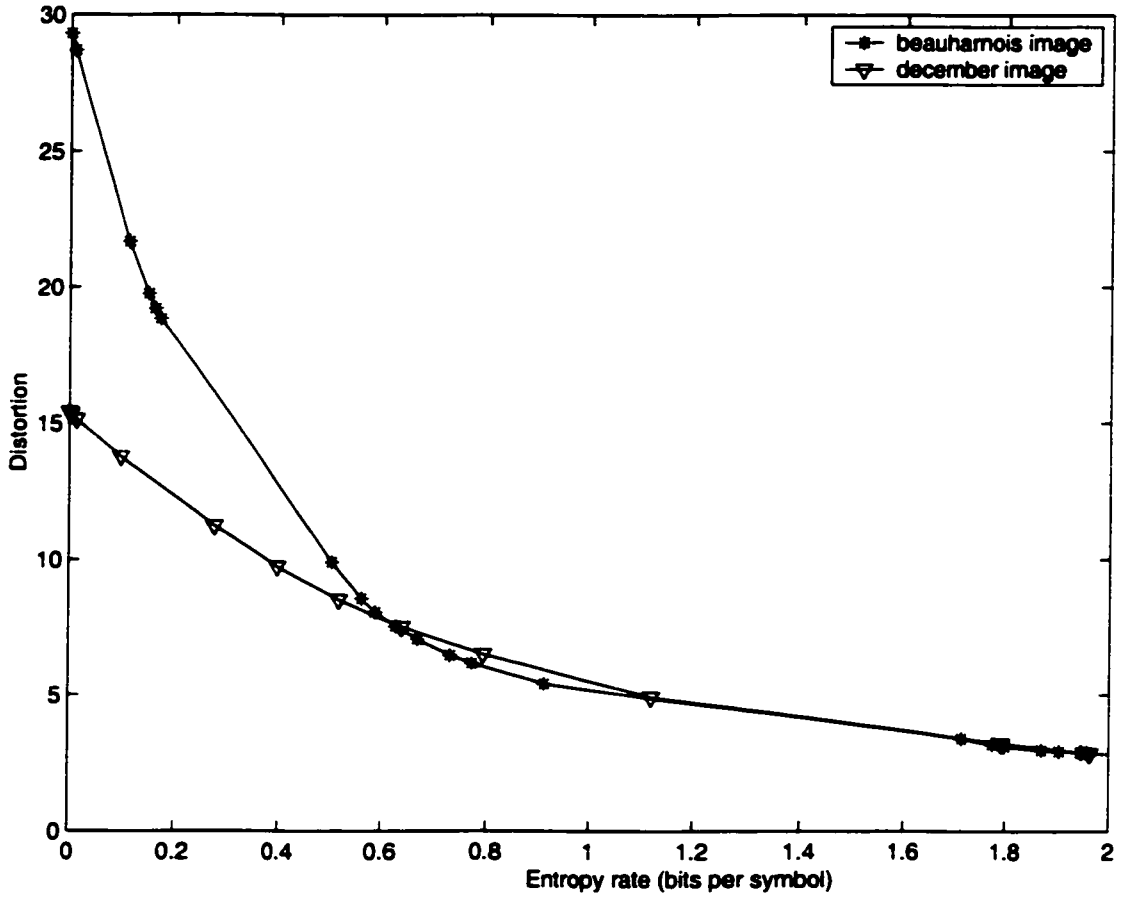

Figure 3.1: Performance of the ECVQ algorithm on quantization of chrominance information of the beauharnois and the december images. The size of the initial codebook is 8.

the typical performance of this algorithm when performed on image data (we use the chrominance vectors of the image as explained later in Section 4.3).

It is interesting to note that to obtain points more or less linearly separated on the curves, we use an exponential progression of the Lagrangian multiplier λ, driven by two parameters that allow us to change the number of points on the curves and the shape of the exponential progression, from almost linear to really curved, in order to better match the distortion–rate curves. Other techniques were suggested in [6] for use with the ECVQ algorithm. We tried them with little success. We also designed

a dichotomous algorithm, but it proved to be much slower than the exponential progression of λ. The former one will also be used for the CECVQ algorithm.

Theoretical Analysis

We do not claim to redemonstrate what was already done by Chou and Lookabaugh in [6], but instead give a minimal understanding of the algorithm design and functionalities. We will use this analysis in Section 3.2.3 to better understand the CECVQ algorithm.

If we reuse the notations of Chou and Lookabaugh in [6], the ECVQ can be seen as a n-iterative transformation T of three functions $\alpha^{(t)}$, $\gamma^{(t)}$ and $\beta^{(t)}$, so that the objective function $J_\lambda(\alpha^{(t)}, \gamma^{(t)}, \beta^{(t)})$ is a decreasing function of the iteration index t.

The function α (Step 1) represents the search of the nearest codeword for every input vector, in the sense of the objective function J_λ. The function γ (Step 2) represents the recalculation of the statistical properties (the codeword probabilities in the ECVQ). The new entropy is a decreasing function because of the constraint on the entropy in the function α . Then, the function γ makes the entropy part of J decrease. The function β (Step 3) represents the move of the codewords to the centroids. The function β makes the distortion part of J decrease in the same way as the GLA.

We have divided each iteration in three phases (or functions). Each phase makes the objective function decrease. Thus, we ensure a global decrease. This process can be summarized by the following equation:

$$J_\lambda(\alpha^{(t+1)}, \gamma^{(t+1)}, \beta^{(t+1)}) \leq J_\lambda(\alpha^{(t+1)}, \gamma^{(t+1)}, \beta^{(t)}) \leq J_\lambda(\alpha^{(t+1)}, \gamma^{(t)}, \beta^{(t)}) \leq J_\lambda(\alpha^{(t)}, \gamma^{(t)}, \beta^{(t)})$$

We concentrate now on the function α and the process that ensures that the objective function is going to decrease from iteration t, Step 3 to iteration $t+1$, Step 1. The question is: "How can we be sure that the new mapping designed by the function α will make the function J decrease?". In other words, can we be sure that the choice of the best codeword for every input vector will decrease J from one iteration to the next?

For the ECVQ algorithm, the answer is fairly simple. From iteration t, Step 3 to iteration $t+1$, Step 1, we know that the statistical measures (function γ) and the codebook (function β) are fixed, which means that for an input vector x and its associated codeword c_x at the α phase we have $d^{(t+1)}(x, c_x^{(t)}) = d^{(t)}(x, c_x^{(t)})$ and $P^{(t+1)}(c_x^{(t)}) = P^{(t)}(c_x^{(t)})$. It is then obvious that $j^{(t+1)}(x, c_x^{(t)}) = j^{(t)}(x, c_x^{(t)})$. Now, either a better codeword exists and we have $j^{(t+1)}(x, c_x^{(t+1)}) < j^{(t)}(x, c_x^{(t)})$ with $c_x^{(t+1)} \neq c_x^{(t)}$, or we choose the same and we have $j^{(t+1)}(x, c_x^{(t+1)}) = j^{(t)}(x, c_x^{(t)})$ with $c_x^{(t+1)} = c_x^{(t)}$. When we add the partial functions for each input vector x, we have $J_\lambda(\alpha^{(t+1)}, \gamma^{(t)}, \beta^{(t)}) \leq J_\lambda(\alpha^{(t)}, \gamma^{(t)}, \beta^{(t)})$ as stated before.

3.2.2 Conditional Entropy-Constrained Vector Quantization

Now, we can also consider entropy of higher orders. In this case we talk about Conditional Entropy, that is conditional to a neighborhood, whose size can be 1,2, 4 or more. Apart from the calculation of the entropy, the design of the CECVQ algorithm is exactly the same as the ECVQ algorithm described in Section 3.2.1. The CECVQ algorithm was first derived from the ECVQ algorithm by Chou and Lookabaugh in [5].

In the computation of the objective function between an input vector and a given codeword, the entropy is calculated according to a chosen probabilistic model. The use of the full model is not possible in practice due to the excessive amount of calculation that it would require. Instead, we suppose that the data follows a Markov model. For the computation of a n-order entropy we use a n-order Markov model. This changes the way the probability of the codewords is computed: instead of $P(c)$, we use $P(c|\mathcal{N}_c)$, where $\mathcal{N}_c$ is the neighborhood of c. Then, the partial objective function is $j(x, c) = d(x, c) - \lambda \log_2 P(c|\mathcal{N}_c)$. To speed up computation, simplify algorithm design and match with the encoder probabilistic model, we can also use a reduced probabilistic model as explained in Section 4.2.2.

Although we have noted that the CECVQ algorithm has the same design as the

ECVQ algorithm apart from the entropy calculation, this difference is a major difference. At the iteration $t+1$, for the computation of the probability of a codeword c, we can decide to use the neighborhood $N_c^{(t)}$ of the iteration t or the neighborhood $N_c^{(t+1)}$ of the present iteration $t+1$, which is different from $N_c^{(t)}$ when one or several codewords have been changed due to a new mapping of previous input vectors. The former implementation will be called CECVQa and the latter implementation CECVQb. We will now study both of these implementations to determine which one is the most suitable for our need.

CECVQa

In this implementation, the neighborhood is taken from the mapping of the previous iteration. We also use the statistical properties that correspond to this mapping, since at the α phase the probabilistic model and the codebook are fixed (see Section 3.2.1). This seems then to be the most legitimate implementation.

Fig. 3.2 shows the distortion–entropy property of this algorithm applied on an image for several codebook sizes and considering two neighbors. The entropy decreases with an almost constant distortion until a certain threshold. Fig. 3.3 shows the codeword map at this threshold for a codebook of size 4. Then, the distortion increases with an almost constant entropy. The experiments showed that after a certain threshold, the codeword map does not really change but undergoes a sort of shifting effect as shown in Fig. 3.4. When the constraint on the entropy increases, the different areas are not reorganized but, instead, show a translation movement, which has for effect to increase the distortion without decreasing the entropy rate. This effect is really not welcomed and makes this part of the curves completely useless.

Finally, only a small part of the distortion–entropy characteristic of the CECVQa algorithm could be used: the part where the entropy decreases with a slow increase of the distortion. It seems that the shifting artifact appears because the algorithm would not let the codebook size decrease. All along the curve, the codebook keeps that same size as the size of the initial codebook. We did not prove it theoretically,

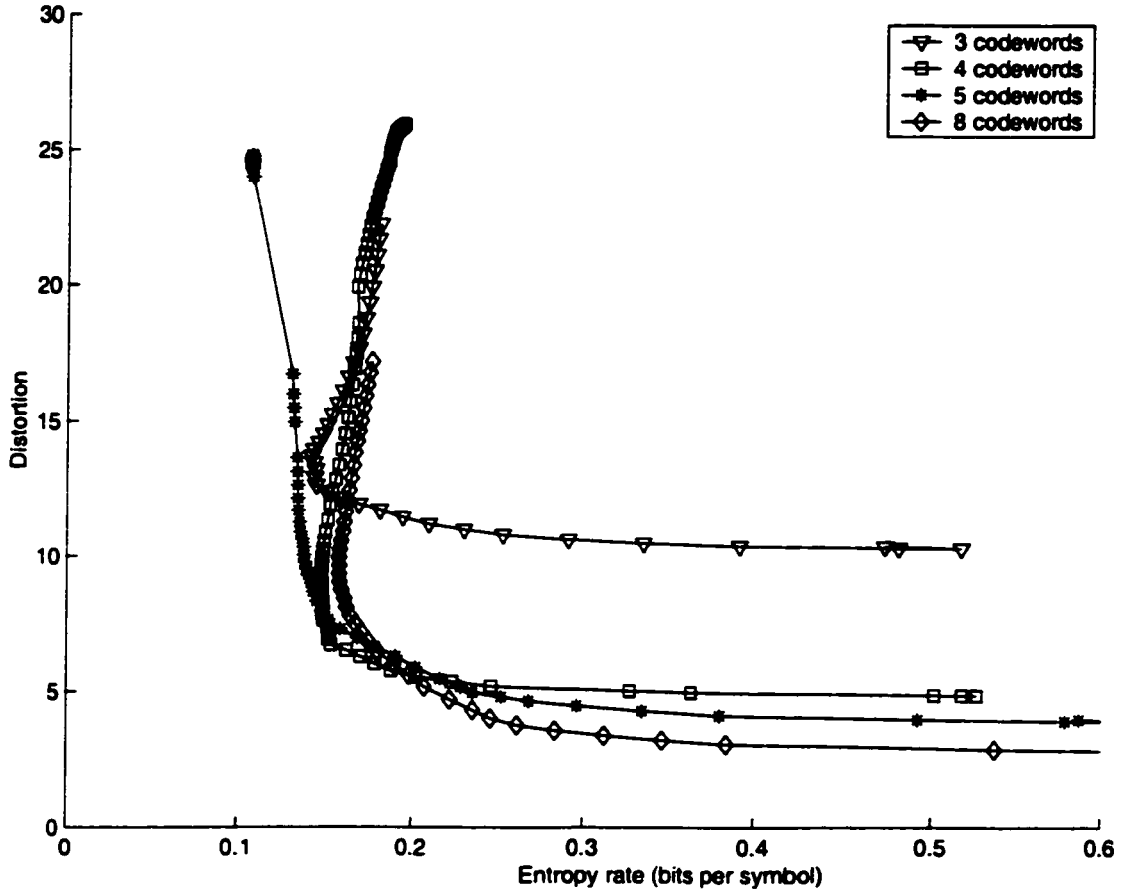

Figure 3.2: Performance of the CECVQa algorithm on quantization of chrominance information of the beauharnois image

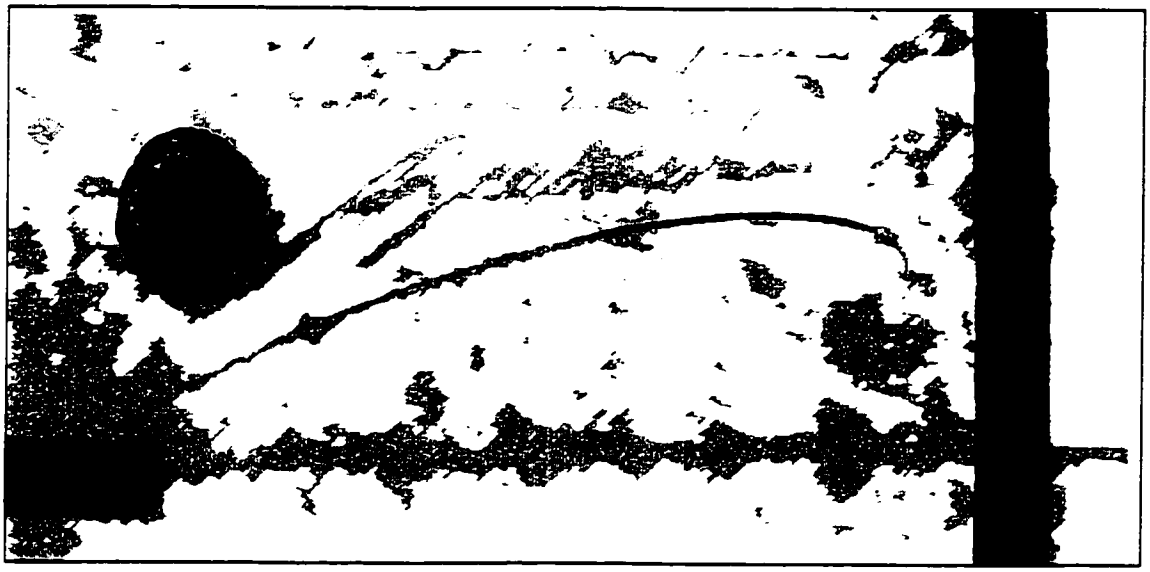

Figure 3.3: Codeword map obtained with the CECVQa algorithm for the beauharnois image. The size of the final codebook is 4.

but all the experiments showed that the codebook would not shrink. Then, the curves cannot reach the point of zero entropy and cannot give a full-ranged curve. This is definitely not what we want.

CECVQb

In the CECVQb algorithm, we use the codewords of the present iteration in the neighborhood. When we consider an input vector x, we compute the partial function $j(x, c)$ for every codeword c in the codebook. The vector x has a certain neighborhood $\mathcal{N}_x$, and each vector of this neighborhood was mapped to a certain codeword just before (we consider the case of a causal neighborhood). The neighborhood $\mathcal{N}_c$ of the examined codeword c is then the current mapping of $\mathcal{N}_x$ and is the same for all the codewords of the codebook. The neighborhood $\mathcal{N}_c$ depends then on x and not on c. From one iteration to another, this neighborhood $\mathcal{N}_c$ associated to an input vector x is changed if one of the neighbors of x changed mapping. This actually occurs quite frequently, especially at lower entropy rate when the constraint on the entropy rate

Figure 3.4: Shifting effect of the CECVQa algorithm on the beauharnois image

is stronger.

Fig. 3.5 shows the distortion–entropy property of the CECVQb algorithm applied on an image for several codebook sizes and considering two neighbors. Contrary to the CECVQa algorithm, we obtain full-ranged curves. The codebook shrinks normally from its initial size to a 1-codeword codebook. First, the entropy decreases while the distortion increases slowly, then the distortion increases quickly to its maximum on a short entropy range. These curves correspond to excellent distortion–entropy performance, since it is possible to attain a very low entropy rate for a minimum distortion. Compared to the performance of the ECVQ algorithm shown in Fig. 3.1, we see the benefit of higher-order entropy calculation. The ECVQ algorithm cannot reach such low entropy rates while keeping a low distortion.

However, the CECVQb algorithm does not follow the requirements of descent iterative algorithm. When the constraint on the entropy increases, it loses its descent property. For high values of λ, the objective function does not decrease but instead increases or shows an oscillatory phenomenon from one iteration to the next.

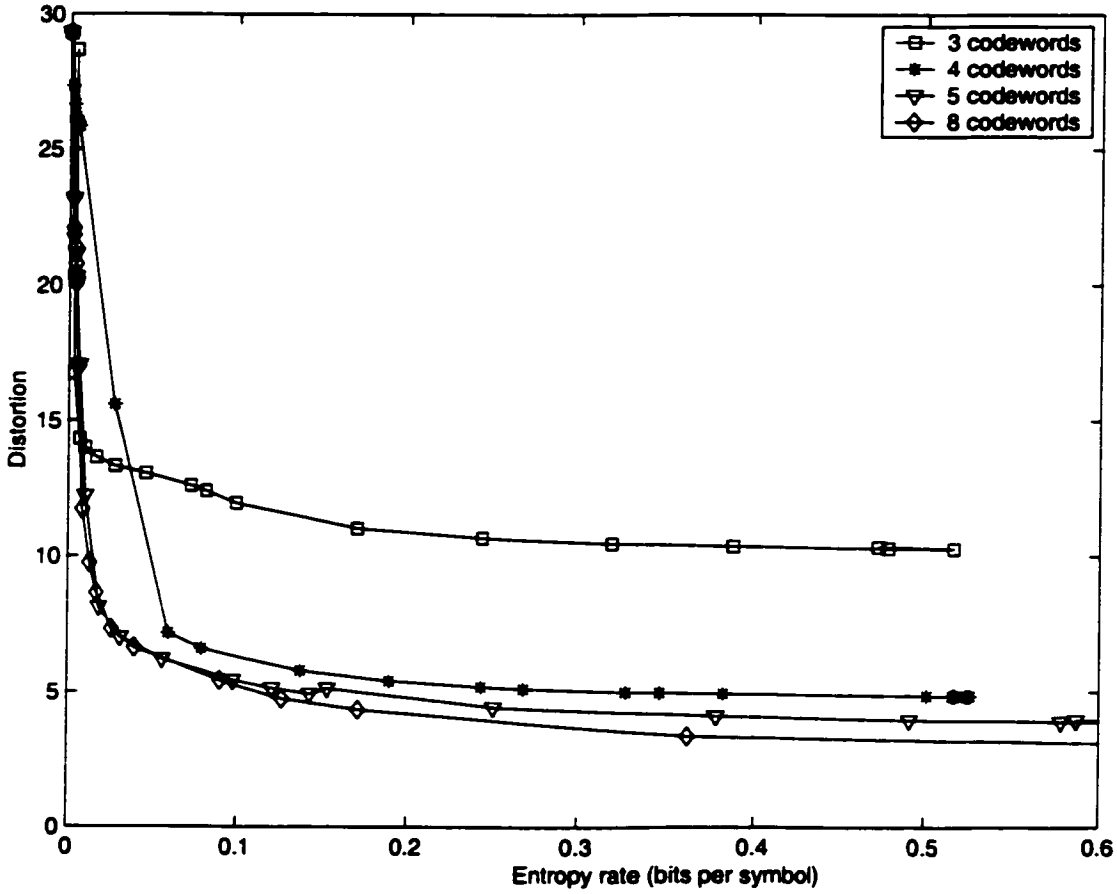

Figure 3.5: Performance of the CECVQb algorithm on quantization of chrominance information of the beauharnois image. Each curve corresponds to a different size of the initial codebook.

3.2.3 Improvements to CECVQ for image quantization

Even if the CECVQ algorithm is theoretically very similar to the ECVQ algorithm, the adaptation of the ECVQ algorithm to the CECVQ algorithm appears not to be as straightforward as we thought.

The CECVQa algorithm prevents the codebook from shrinking when the constraint on the distortion increases. Then, the curve reaches a minimum rate value for each size of the codebook, but the distortion keeps increasing when the constraint on the rate increases. Visually, we see a curious translation of the codeword map image. This effect is not suitable at all, since we do not know *a priori* the size of the final codebook, but we prefer to give a target rate to achieve along this curve.

The CECVQb algorithm is not a descent (convergent) algorithm. When an arbitrary number of iterations are run for a given value of λ, the objective function might begin to oscillate between 2 or 3 values. Eventually, it will converge to a certain value, but the intermediate values do not follow any special pattern. The objective function might increase or decrease. The convergent value might be greater than the initial value. The algorithm does not minimized the objective function. But, on the other hand, it shows the distortion–entropy performance we are looking for: the possibility to achieve low entropy rate with minimum distortion increase.

CECVQm

It seems legitimate to think that if we can modify the CECVQb algorithm to make it a descent algorithm we can expect to achieve better distortion–rate performance. For that, we need to better understand why the CECVQb algorithm is not necessarily convergent. Since, the only difference between the ECVQ algorithm and the CECVQ algorithm is the calculation of the entropy, which includes inter-codeword information in the case of the CECVQb algorithm, we will try to understand what is happening in the latter case.

In the case of CECVQ, we have $j(x,c) = d(x,c) + \lambda h(c|\mathcal{N}_c)$, with $h(c|\mathcal{N}_c) = -\log_2 P(c|\mathcal{N}_c)$. We consider the same reasoning as for the ECVQ in Section 3.2.1.

From iteration t, step 3 to iteration $t+1$, step 1 and with a codeword c_x mapped to an input vector x, we have $d^{(t+1)}(x, c_x^{(t)}) = d^{(t)}(x, c_x^{(t)})$ and $P^{(t+1)}(c_x^{(t)}) = P^{(t)}(c_x^{(t)})$. However, we do not have necessarily $\mathcal{N}_c^{(t+1)} = \mathcal{N}_c^{(t)}$, because the neighborhood might have been changed if a previous input vector was mapped to a codeword different from the one of the previous iteration.

Then, in general, we have $P^{(t+1)}(c_x^{(t)}|\mathcal{N}_c^{(t+1)}) \neq P^{(t)}(c_x^{(t)}|\mathcal{N}_c^{(t)})$. And if we map x to the same c_x as in the previous iteration, $h(c_x|\mathcal{N}_{c_x})$ might increase since the neighborhood might have been changed. The condition $j^{(t+1)} \leq j^{(t)}$ is not true anymore and the algorithm is not necessarily a descent algorithm.

We want to modify the CECVQ in order to keep the descent properties. We suggest that in the α function, we associate the input vector to the codeword that minimizes this modified partial objective function:

$$\overline{j(x,c)} = d(x,c) + \lambda \left(h(c|\mathcal{N}_c) + \sum_{c_i \in \mathcal{N}_c} h(c_i|\mathcal{N}_{c_i}) \right)$$

This means that all the neighborhoods to which the inspected codeword c belongs are taken into account, so that the modification of these neighborhoods will not result in an increase of the objective function when they are inspected in their turn. We call CECVQm this modified implementation.

The CECVQm algorithm proves theoretically and experimentally to be a descent algorithm. It is decreasing in $\overline{J_\lambda}$. But it can also easily be shown that $\overline{J_\lambda} = J_{|\mathcal{N}|\cdot\lambda}$, since $\sum_c \left(h(c|\mathcal{N}_c) + \sum_{c_i \in \mathcal{N}_c} h(c_i|\mathcal{N}_{c_i}) \right) = |\mathcal{N}| \cdot \sum_c h(c|\mathcal{N}_c) = |\mathcal{N}| \cdot H$ ($|\mathcal{N}|$ is the cardinality of $\mathcal{N}$). The algorithm is then decreasing in J_λ as well.

Fig. 3.6 shows the characteristic of the CECVQm algorithm. Surprisingly, the curves are not full-ranged. Except when starting with a codebook of size two, the entropy decreases with a low increase of the distortion and converges to a certain distortion–entropy point. No matter how high is the constraint on the entropy, we cannot go further than this point. One more time, the codebook size does not change. Moreover, for almost every size of the initial codebook, the CECVQm algorithm is outperformed by the non-descent CECVQb algorithm.

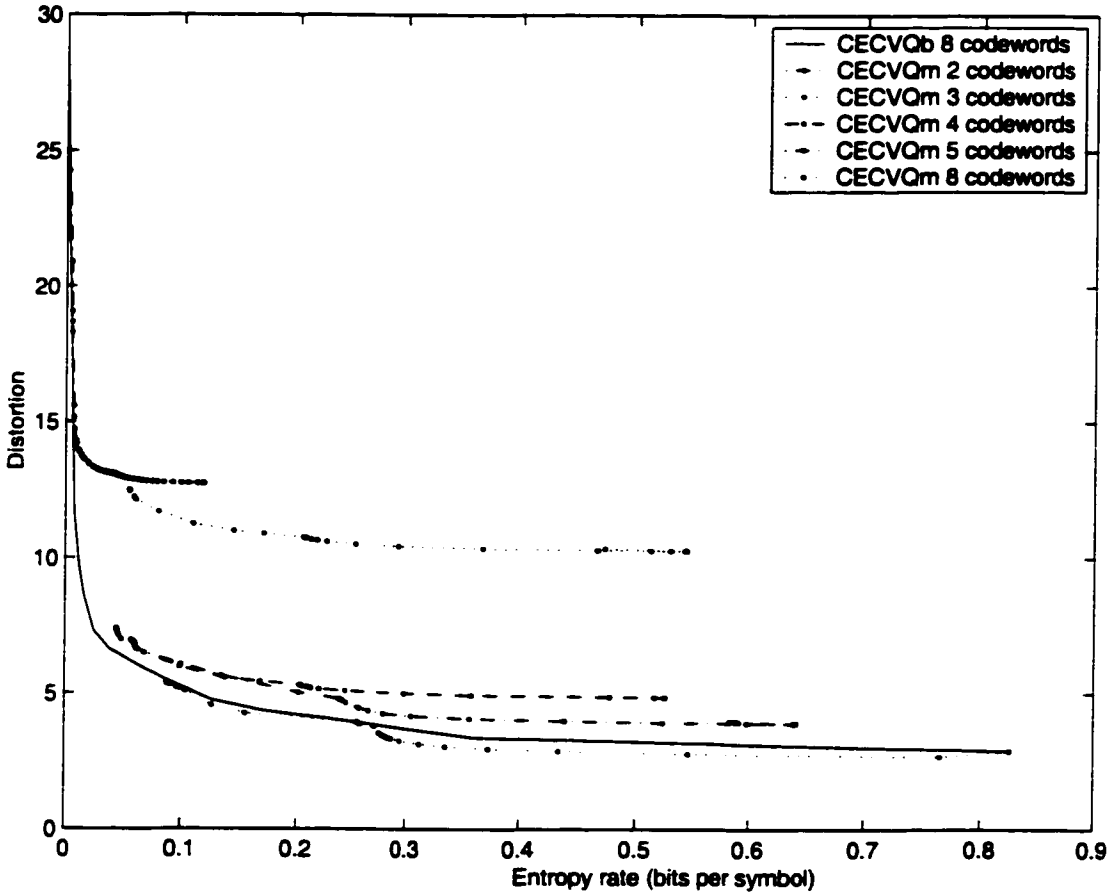

Figure 3.6: Performance of the CECVQm algorithm on quantization of chrominance information of the beauharnois image. The neighborhood size is 2 and the CECVQb algorithm is the reference. Each curve corresponds to a different size of the initial codebook.

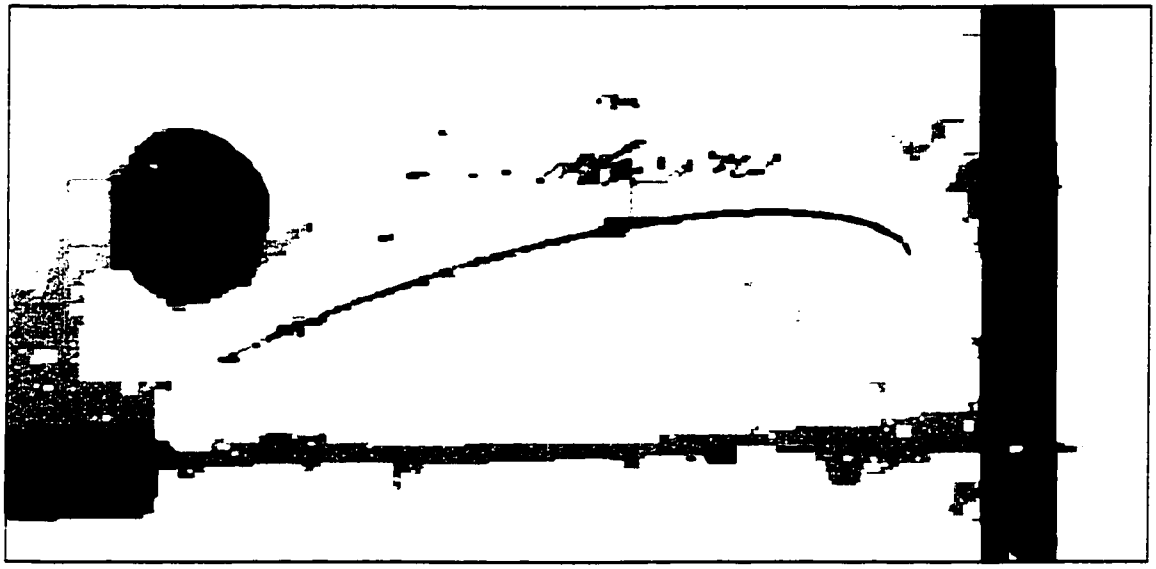

Figure 3.7: Example of a final codeword map obtained with CECVQm for a maximal constraint on the entropy rate. The size of the final codebook is 4 and the neighborhood size is 2.

In the limit, when the λ multiplier reaches very high values, we obtain the typical codeword map shown in Fig 3.7. We note some strange small monodimensional structures more or less oriented in the direction of the neighborhood. It looks like a cluster tries to grow over another from both sides but can never join, because of the anticipation we included in our partial objective function. However, we were not able to fully explain this interesting phenomenon.

CECVQf

We suggest to finish the following algorithm: we run one iteration of the CECVQb algorithm to ensure the shrinking of the codebook if required and the move of the codebook to another distortion–rate point of the curve. Then, we optimize the distortion–rate tradeoff with the CECVQm algorithm. The complete algorithm will be called CECVQf. The total number of iterations depends on the chosen threshold, and more generally, the convergence speed depends on the size of the codebook (the bigger the codebook, the slower the convergence). By doing so, we respect the two conditions

we needed. First, we can obtain a distortion–rate curve ranging from a null rate to a null distortion. And second, we have a descent algorithm that allows to optimize little by little the curve through iterations. We will now consider the property of this curve that validates the performance of the designed algorithm.

It has to be noted that these are the results we obtained using a greedy algorithm. In [5], Chou and Lookabaugh also suggest the use of dynamic programming to find the codeword sequence that minimized the objective function. This has yet to be done for 2-dimensional problems such as images.

3.3 Distortion–rate performance

The CECVQf algorithm yields very good distortion–rate performance. As expected, it performs much better than the ECVQ algorithm, since spatial information is considered.

We consider the performance of the CECVQf algorithm according to its response to the neighborhood size and the initial codebook size.

Fig. 3.8 shows the influence of the neighborhood size on the algorithm performance. A neighborhood with only one neighbor is clearly less efficient, and this is not surprising since we are dealing with images. On the other hand, a neighborhood bigger than 2 does not seem to bring much better performance. A neighborhood of size 2 is then the best tradeoff between the gain in performance and the additional time spent. We also notice that a reduced probabilistic model (as explained in Section 4.2.2) denoted as "cecvq-f2-red" in Fig. 3.8 performs similarly to a full probabilistic model for a neighborhood size of 2. All algorithms were run on an initial codebook of size 4.

Fig. 3.9 shows the performance of the algorithm as a function of the initial codebook size. It appears clearly that a larger initial codebook is better. It would perform better at high rate and possibly at low rate as well. However, when the initial codebook gets larger, the algorithm design time increases. It can then be advantageous

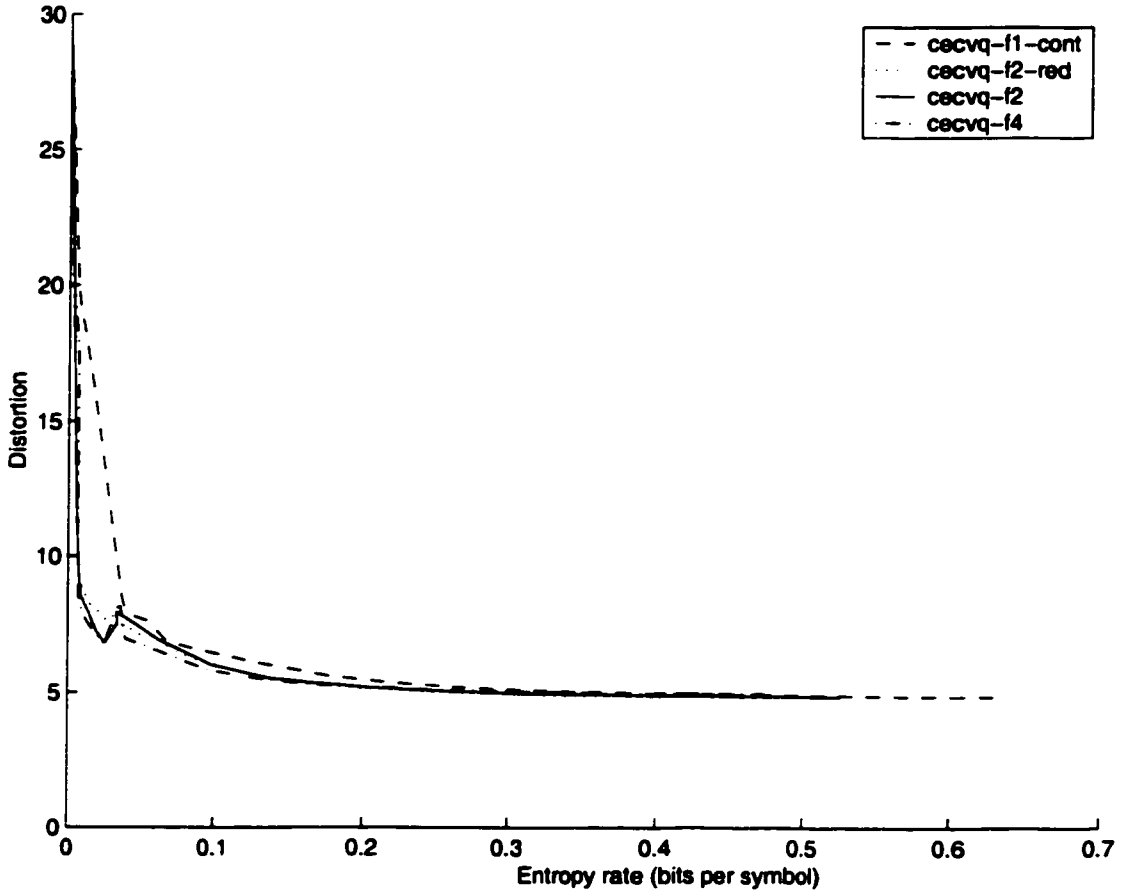

Figure 3.8: Influence of the neighborhood on the CECVQf algorithm with the beauharnois image. The neighborhood size is indicated by the name of the algorithm: "cecvq-f2" indicates a neighborhood size of 2.

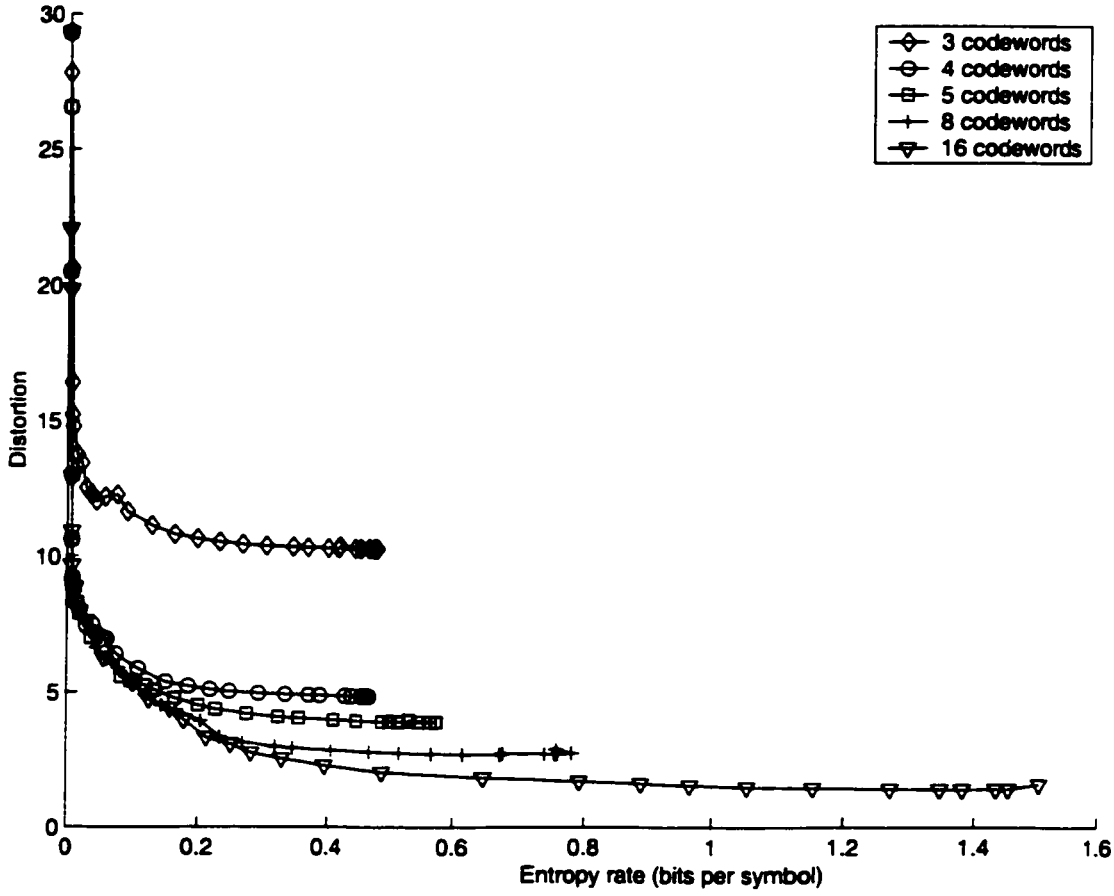

Figure 3.9: Initial codebook size influence on the CECVQf algorithm performance

to have an idea *a priori* on the ideal size of the initial codebook.

A drawback of this algorithm is the difficulty to achieve a low entropy rate easily. We must start with high rate and decrease it little by little until the desired value of the entropy rate. The weakness of the scheme is in the dependence of the achieved distortion–rate point on the previous points. The less we use intermediate points to go to our target rate, the more likely it is that we will be stuck in another local minimum, far less optimal. The best strategy we developed so far, is to increase λ exponentially. The exponential increase is controlled by two factors: one determines the number of points on the curve and the other controls how concave the curve is (that is the distribution of the points between low values and high values). We noted that linear and dichotomous methods were far less efficient in term of distortion–rate performance as well as for the computing time. The difficulty of obtaining a given entropy rate in a straightforward way remains the main weakness of this algorithm.

Another drawback is that the CECVQf does not perform much better than the CECVQb and can sometimes even be worse! The CECVQf does only one iteration of CECVQb, against at least two for CECVQb. Then CECVQf performs distortion minimization through CECVQm. Fig. 3.10 shows the influence of the number of CECVQm iterations on the minimization of the objective function. The number of iterations is determined by the threshold level. For a threshold of 1, only 2 iterations of CECVQm are done. A lower threshold leads to more CECVQm iterations. However, the additional reduction in distortion proves to be small as shown in Fig. 3.10. The performance improvement with a very high number of CECVQm iterations is almost null! The points obtained on the distortion–rate curves are not the same, but the curves are the same.

3.4 Related algorithms

In the same area of research, that is, on conditional entropy-constrained vector quantization, other algorithms were suggested and are worth a word about it.

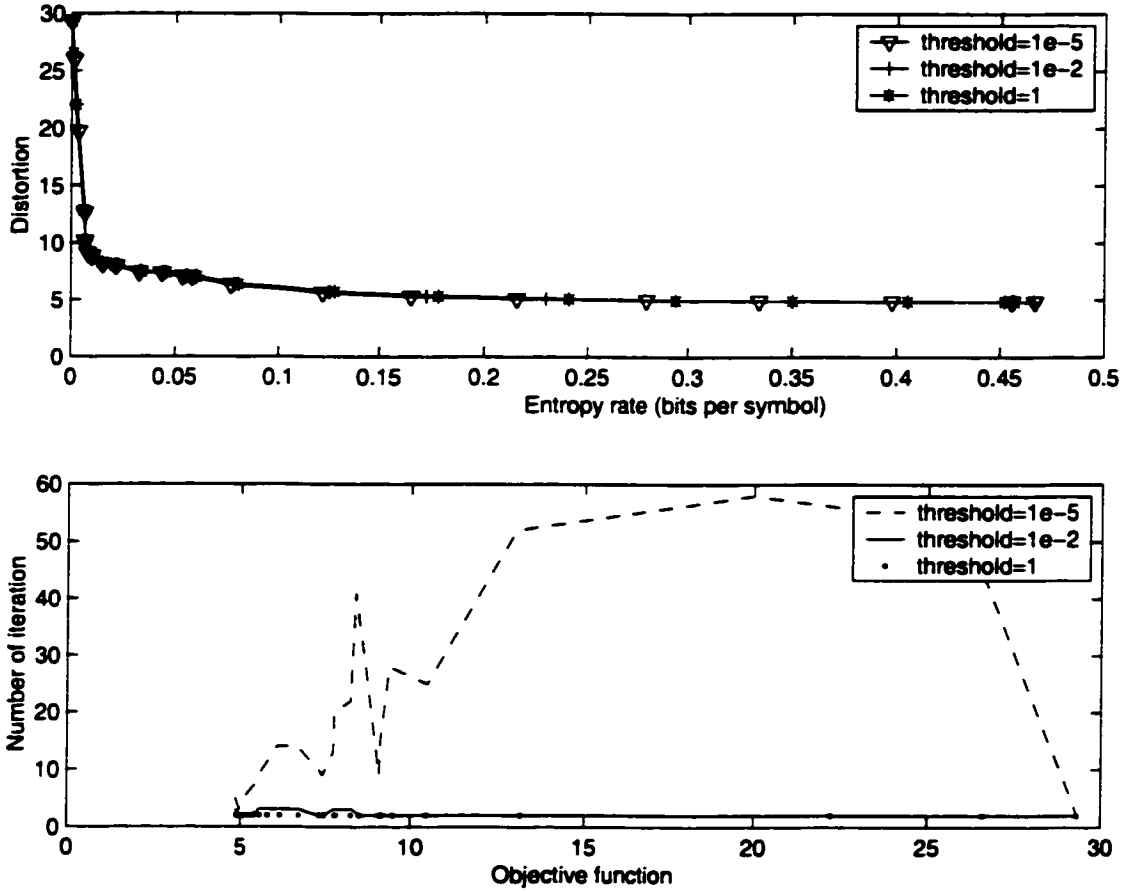

Figure 3.10: Influence of the threshold value on the CECVQf algorithm performance

3.4.1 Conditional Entropy-Constrained Pairwise Nearest Neighbor

In [17], W. Pearlman studies high-rate theory and conditional entropy-constrained vector quantization and proposes a new algorithm: the Conditional Entropy-Constrained Pairwise Nearest Neighbor (CECPNN) algorithm. The CECPNN is an algorithm that designs codebooks by merging the pair of Voronoi regions which gives the least increase of distortion for a given decrease in entropy of first order or higher.

The approach is then completely different from the original CECVQ algorithm, since the CECPNN actually reduces the codebook iteration by iteration until the required size. A codebook large enough must then be initially designed. Also, the algorithm was developed on the basis of high-rate theory. But, here, we are especially interested in low rates where the CECPNN does not make much difference with the CECVQ. However, the CECPNN algorithm could be used for oversegmentation reduction, for example after a watershed segmentation, as suggested in Section 4.7.2.

3.4.2 Conditional Entropy-Constrained Tree-Structured Vector Quantization

In [18], W. Pearlman proposes an adaptation of the CECVQ algorithm to tree-structure vector quantization, called Conditional Entropy-Constrained Tree-Structured Vector Quantization (CECTSVQ) and applies it to synthetic sources and speech data. The root of the tree is set to the whole data. Then, at each iteration, each new leaf is examined and binary CECVQ algorithm is performed on them. Among the whole set of leaves, the leaf that offers the best distortion–rate tradeoff is split. The process is repeated until the desired codebook size is reached. Each leaf represents a codeword. Pearlman shows that the CECTSVQ algorithm does not perform much worse than the CECVQ algorithm, whereas it is much faster.

This algorithm has still to be implemented for images. Our attempt could not

give a proper segmentation of the images and showed convergence problems. However, we think that the CECVQm algorithm we discussed previously in Section 3.2.3 could be a good candidate for leaf splitting. The CECVQm algorithm with only two codewords, as needed for leaf splitting, shows a complete, almost ideal characteristic (first, the rate decreases with a constant distortion, then, the distortion increases until its maximum at the smallest values of the rate), as shown in Fig. 3.6.

Chapter 4

A New Method: Conditional Entropy-Constrained Coding of Chromatic Information

Taking advantage of the superior performance of the CECVQ algorithm we try to define an image compression scheme especially adapted to the compression of archive document images.

4.1 Method description

4.1.1 Principle

To overcome the difficulties of the recent formats such as JPEG 2000. DjVu and LDF that we pointed out in Chapter 2, we suggest a different approach to encode document images, based on the development of the CECVQ algorithm that we presented in Chapter 3.

Recent work on color spaces [10] and color vector quantization [9] suggest the

possibility of exploiting the color characteristics of the document images described in Section 2.1. In an appropriate color space in which the color information – also called chrominance information – is not correlated with the luminance information. we expect the color components to follow a piecewise-constant model with a limited number of values. This approach is based on the feeling that a color mainly depends on the physical properties of an often homogenous surface, whereas luminance depends on lights and shades. Under this assumption, the chrominance space can be divided into almost constant color areas. Each area can then be represented with a unique value.

This hypothesis combined with the low number of colors seen in archive document images led us to consider the use of vector quantization to separate the different colors. Each color of the 2D chrominance space is mapped to a unique codeword. We obtain a label image that is then encoded with an efficient encoder.

4.1.2 Method

Our new method has three main steps. which are the color space transform, vector quantization of the chrominance information and encoding of the resulting label image:

Step 1: An image to be encoded is first transformed into another color space, in which luminance information and color information are decorrelated. A typical color space that can be used is the YCbCr color space. The color information is represented by a 2-dimensional space (CbCr) and is called chrominance information.

Step 2: The chrominance information is quantized with the CECVQ algorithm developed in Chapter 3. Typically, 3 to 8 levels of quantization are used. each one theoretically representing a different color. However, the main criterion remains the target entropy rate. which determines the final size of the codebook. The codeword map corresponding to this quantization is called the label image

and has an entropy rate close to the target entropy rate of the CECVQ. The chrominance information was then turned into a label image and a codebook.

Step 3: The label image is encoded with an entropy coder that can achieve a rate close to the entropy rate of the label image. The codebook is usually encoded in the same file but contributes a negligible part of the file size for typical codebooks and images.

The luminance information can be either encoded in the same way as the chrominance information or using wavelet transforms as in JPEG 2000.

4.2 Relation between label map and arithmetic coder

4.2.1 Label map

In the label map, each label is an index into a table where the quantized chrominance vector can be found. According to the piecewise-constant model of the chrominance space, the label map should define homogenous areas. Then, each label can be seen as a color and refers to a couple of values (Cb, Cr). Homogenous areas correspond to objects or homogenous surfaces, such as the paper, the ink or a seal. Fig. 4.1 shows a typical label image corresponding to the quantization of a piece of an archive document image.

Zero-order entropy depends only on label probabilities. But higher-order entropies depend on label probabilities and on inter-label information. This inter-label information is usually measured by conditional probabilities, which are the probability of a label given its neighborhood. A n-order entropy calculation uses a neighborhood of size n and n-order entropy rate value is always less than or equal to n-1-order entropy rate value.

The entropy value (other than zero-order entropy) depends on the importance of inter-label information. The more inter-label information there is in the image, the

Figure 4.1: Example of a 4-codeword label map for the beauharnois image (compression ratio 289:1)

lower is the entropy. Homogenous areas correspond to areas where labels are highly correlated. Noisy areas are areas where labels are loosely correlated. Then, to have low entropy, the areas must be as homogenous as possible.

4.2.2 Context-based arithmetic coder

An arithmetic coder tries to match its rate to the entropy rate calculated with the probabilistic model given to it. However, to be able to decode, an arithmetic coder needs also to encode the probabilistic model of the data with the encoded data. The more complex this model is, the lower the attainable rate is, but also the greater are the number of bits required to store the probabilistic model. Since the use of a bigger context (or neighborhood) does not always improve the compression ratio. A tradeoff between the number of bits allocated to the data and the number of bits allocated to the model has to be found to find the best compression ratio achievable.

In the arithmetic coder developed by Lauzon [13], a context-based approach is used. A neighborhood of 2, 4 up to 8 pixels can be considered. However, for a

neighborhood of size K and N labels, the required number of parameters to describe the probabilistic model would be $N^K(N-1)$, which increases exponentially with K. In [13], a new method was introduced to reduce the number of parameters. It is based on a simplification of the probabilistic model. Instead of using the complete model, each label has three occurrences depending on a simplified neighborhood model:

- All the labels of the neighborhood are identical. Then the probability that the examined label has the same value as the labels of the neighborhood is very high.
- The neighborhood is divided in two equal sets of labels with two different values. The probability that the examined label value is one of the two is high.
- All the other possibilities can be represented by a single probabilistic value or by a histogram.

Such a simplification of the context makes the representation of the probabilistic model much less expensive in term of bits, with a penalty in terms of compression of the data that is largely compensated by the higher context-order that can be used.

4.2.3 Using the arithmetic coder on label images

This coder is then suitable to encode the kind of label map we are expecting, that is with very homogenous areas. This context-based arithmetic coder has been proven in [13] to be at least 10-25% more efficient than the most well-known compression schemes of this kind such as PNG, SPIHT, LOCO-I, JBIG, and Lempel-Ziv for this type of label map. However, this coder is not intended to be used with label maps without smooth areas, since the efficiency of the coder quickly drops when the areas are not homogenous any more.

In order to use this very efficient coder, we want to generate label maps with homogenous areas. The piecewise-constant model predicts that it should be easily feasible. However, the numerous nonlinearities in the acquisition process makes it

very difficult, first, to move to a color space in which the chrominance is rigorously independent of the luminance, and second, to identify a single color because of the noise that has been added.

Thus, we use a vector quantization algorithm that should enable us to find the original color in the YCbCr common color space with minimal distortion. In order to obtain smooth areas adapted to the context-based arithmetic coder being used. the quantization is constrained by the value of a conditional entropy.

4.3 Color spaces

4.3.1 YCbCr color space

We said that we have been using the YCbCr color space. We chose this color space because it is the color space usually used (JPEG, JPEG 2000) with the easiest implementation. We could then concentrate on the CECVQ algorithm. However. the chrominance components Cb and Cr are not completely decorrelated from the luminance component, because of numerous nonlinearities involved in the acquisition process. Then, the chrominance image is not exactly discrete and piecewise constant. But. the difference between the different areas are big enough to describe them as almost piecewise constant. The use of CECVQ should then help us to capture the underlying piecewise constant property.

Fig. 4.2 and Fig. 4.3 show the (Cb,Cr) space for two different document images. In Fig. 4.2, we can easily separate three to four different areas by ourselves. But in Fig. 4.3, it seems to have only one area. This is actually coherent with the original images and their decomposition on the YCbCr spaces as shown in the figures of Appendix A. The "beauharnois" image has clearly at least three different colors: the colors of the paper, the seal and the tape. It is quite easy to separate the different areas. But the "december" image does not seem to have many colors. The handwriting itself seems to have taken the color of the background. In this case. the identification of the different color areas is not as easy.

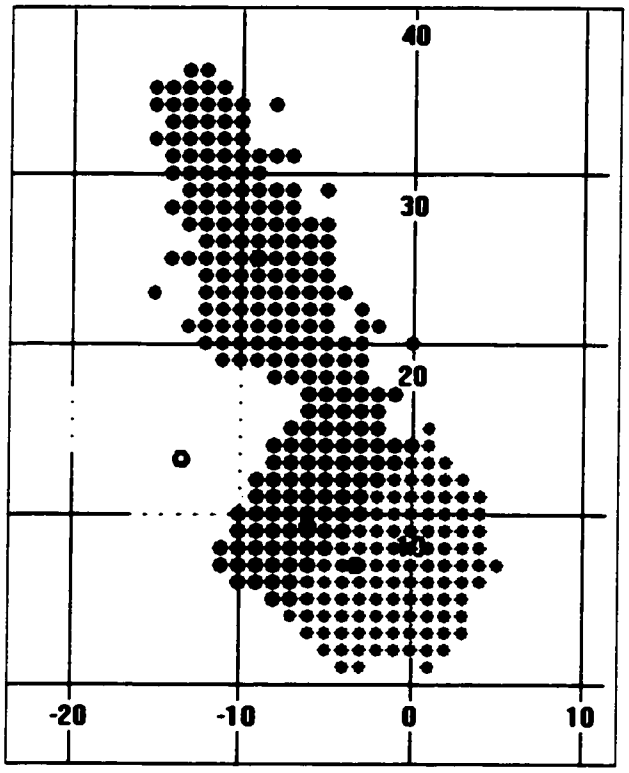

Figure 4.2: Chrominance space (Cr on the vertical axe versus Cb on the horizontal axe) for the beauharnois image (ranges from -128 to 127)

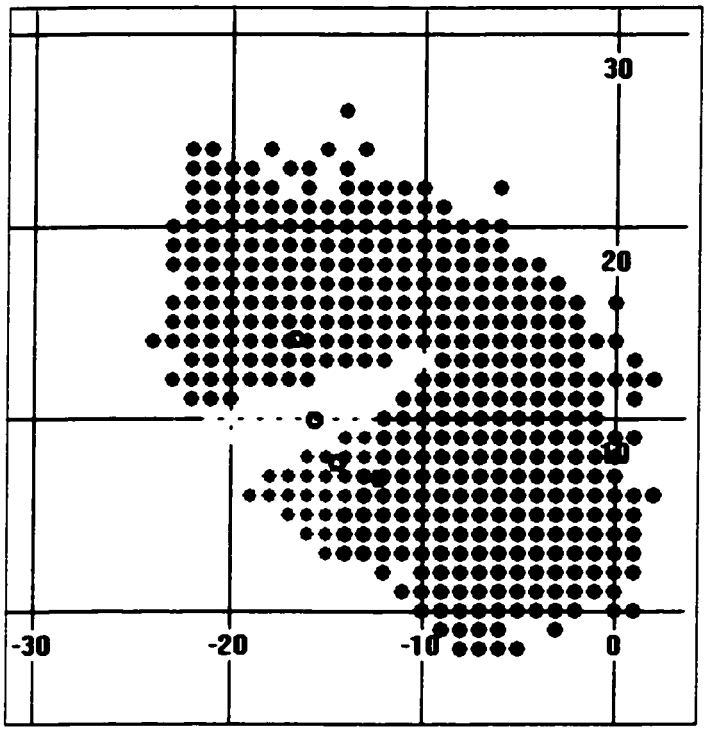

Figure 4.3: Chrominance space (Cr on the vertical axe versus Cb on the horizontal axe) for the december image (ranges from -128 to 127)

4.3.2 Other color spaces

It can be easier to separate the different colors in another color space where the chrominance information is indeed not correlated to the luminance information and presents a stronger piecewise-constant property.

Dubois *et al.* explored in [10] and [9] some alternatives to the YCbCr space. It is suggested to use shifted and scaled chromaticities of the CIELUV color space, which is a perceptually uniform space. The distortion measure used in the CECVQ algorithm would then be directly related to the perceptual distance between two colors. The distortion measure would then not only be a quantitative measure but also a more qualitative measure.

Another suggestion was to estimate the nonlinearity involved in the acquisition process. Dubois *et al.* described in [10] this method called 5-segment estimation. Based on the hypothesis that the chrominance follows the piecewise constant model, a 5-segment estimation of the nonlinearity is performed. The nonlinearity can then be partially compensated and first results showed that images corrected with the estimated inverse nonlinearity are shown to exhibit chromatic properties that are much more piecewise-constant than in the original image.

It can be expected than the use of these color space transforms instead of the RGB to YCbCr transform could lead to enhanced performance. The CECVQ algorithm could more accurately define the edges of each area and give a visually better segmentation of the image. However, because of the complexity and the computing time that these methods add, we chose to use the YCbCr space to be able to fully concentrate on the design of the CECVQ algorithm for images.

4.4 Complete coder

The complete coder is composed as shown in the block diagram of Fig. 4.4. A color space transform is first performed on the original image to isolate the chrominance information (Part A). The 2-dimensional chrominance space is then vector quantized

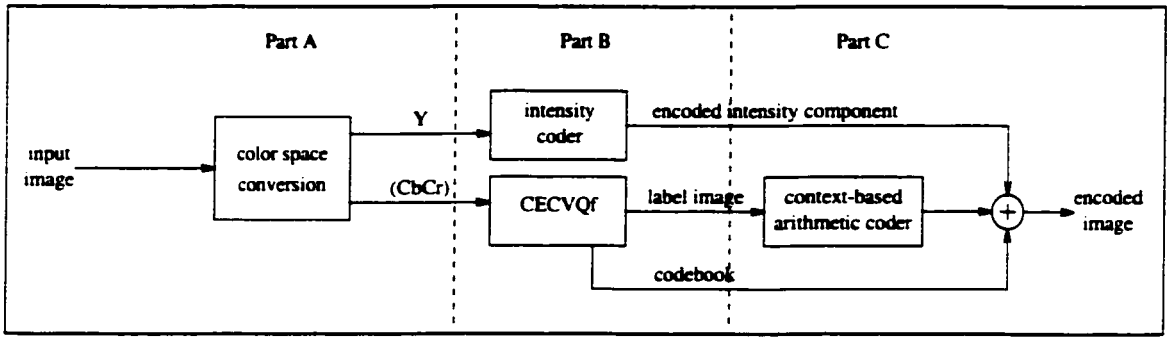

Figure 4.4: Schema of the complete coder

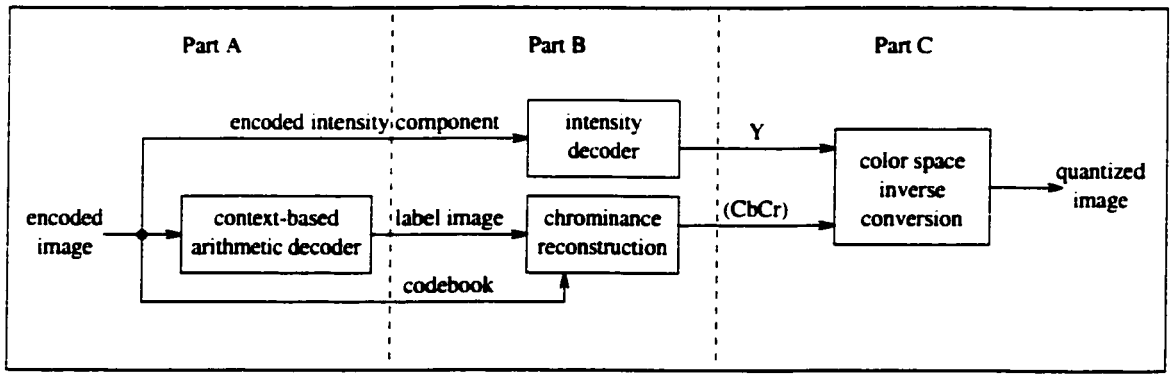

Figure 4.5: Schema of the complete decoder

using the CECVQf algorithm giving a label image and a codebook (Part B). Finally, the label image is encoded with a context based arithmetic coder (Part C). The luminance component can be encoded with another coder, such as the wavelet-based JPEG 2000 coder. The codebook also has to be saved to be able to reconstruct the quantized image.

The decoder does the inverse transformations of the coder. Its architecture is shown in the block diagram of Fig. 4.5. The encoded label image is decoded with the arithmetic coder (Part A). The label image and the codebook are combined together to reconstruct the quantized chrominance information (Part B). After decoding the luminance with the appropriate decoder, it is combined with the chrominance and the inverse color space transform is performed (Part C) to finally obtain the quantized image.

The luminance component left over after the color space transform can be encoded in different ways. For now, we are using a JPEG 2000 encoder to encode it. But it was also thought to run again the CECVQf algorithm in each chrominance cluster to obtain several levels of luminance for each chrominance quantized vector. A single label image would represent this quantization and would be encoded with the context-based arithmetic coder. Even though this method seems promising, we did not have the time to test it and it remains for future study..

4.5 Performance of the method

4.5.1 Distortion–rate performance

The arithmetic coder we used is the coder presented in Section 4.2.2. We used it with a context size of 2. To analyze its performance, we compare its distortion–compression performance to the distortion–entropy characteristic of the label images, which itself corresponds to the performance of our CECVQf algorithm. Fig. 4.6 shows together the distortion–compression curves of the arithmetic coder using a context size of 2 and the distortion–entropy curves of the CECVQf algorithm using a neighborhood of size 2 and the same reduced probabilistic model as the coder. The curves are very close to each other, proving that the performance of the coder is indeed close to its theoretical performance.

The performance of the arithmetic coder is visualized in Fig. 4.6 with distortion–compression curves. However, this compression is expressed in bits per symbol. Since each symbol represents a 2-dimensional vector coded by 8 bits per components, the rate of the arithmetic coder must be divided by 16 to obtain the rate of the chrominance information. A good tradeoff between rate and distortion is the point (rate=0.1 bit/symb, distortion=5.5). This actually corresponds to a compression ratio of 160:1 of the chrominance information and a 240:1 fraction of the total image (0.42%). We can also compress a little more to the point (rate=0.01 bit/symb, distortion=8.9) that corresponds to a compression factor of 1600:1 and 0.042% of the total image

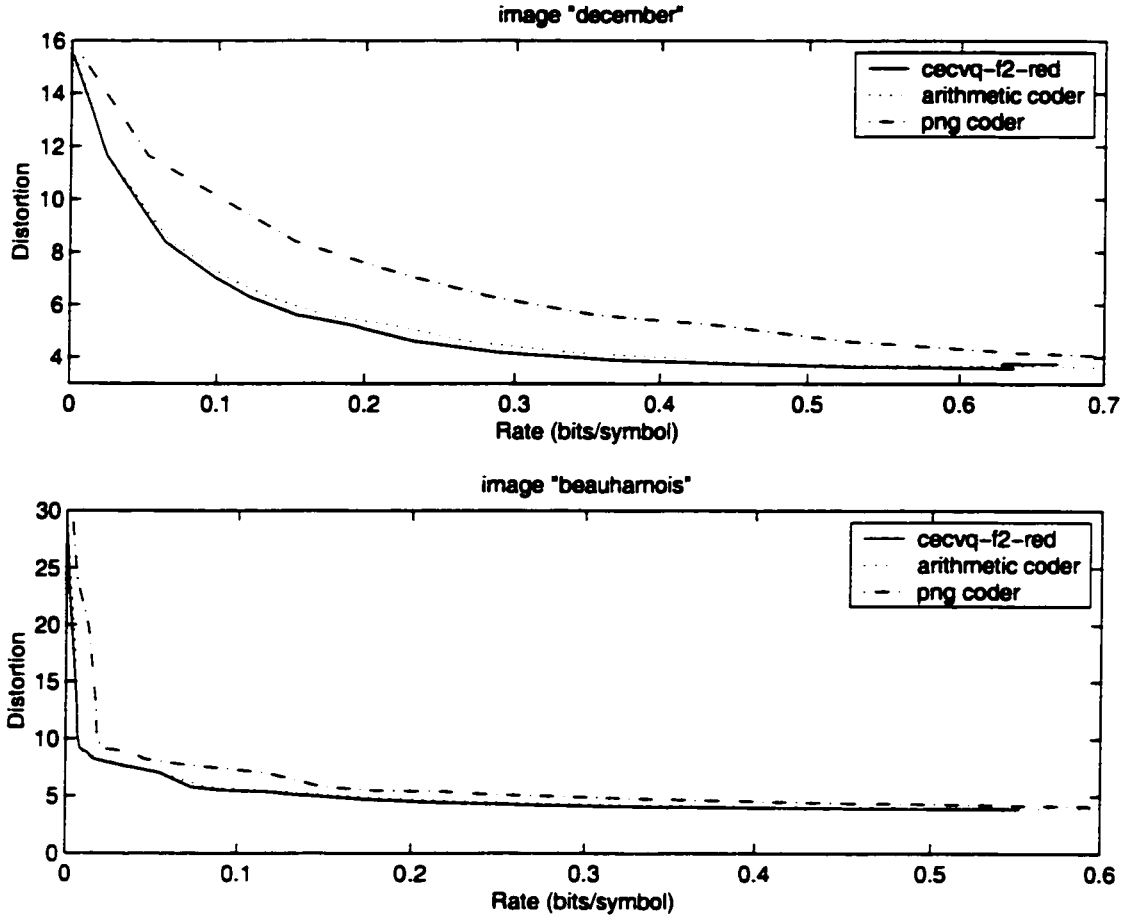

Figure 4.6: Performance of the arithmetic coder and of the PNG coder. The CECVQf characteristic is the reference.

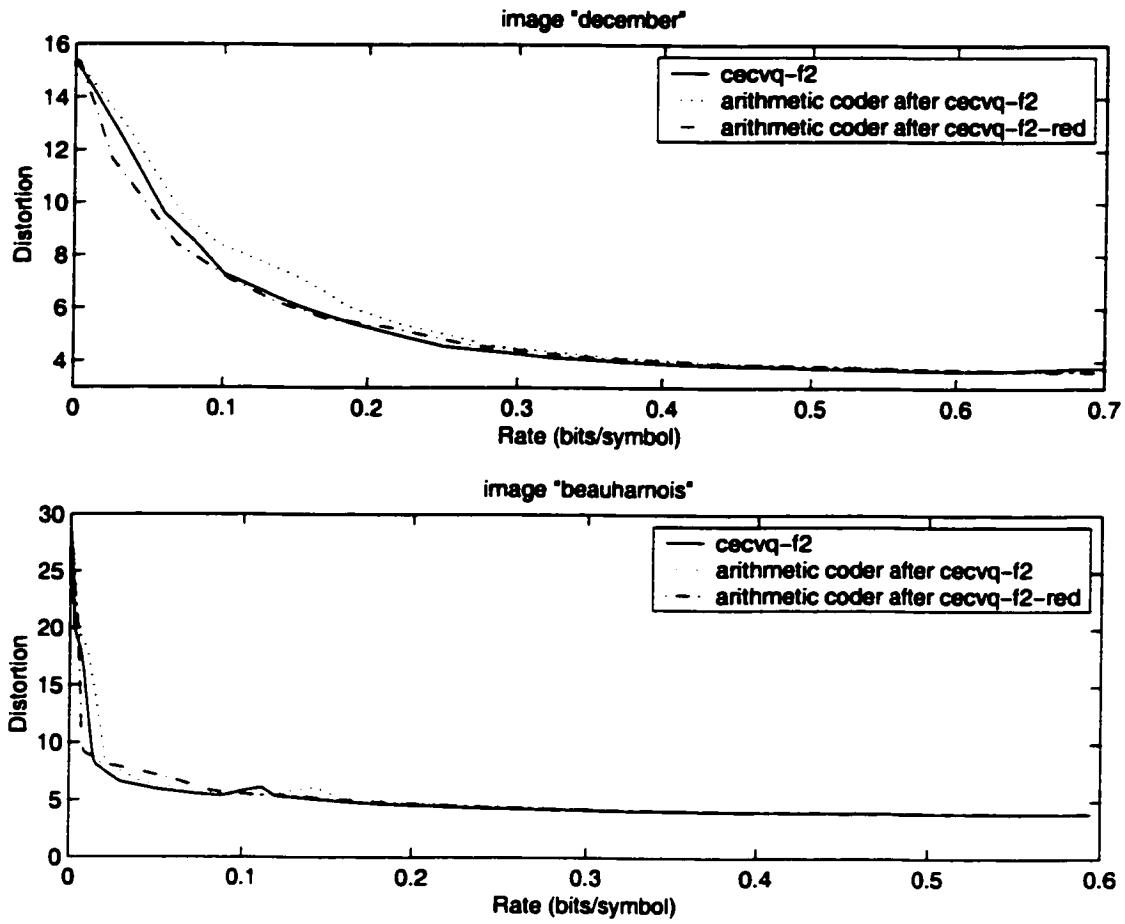

Figure 4.7: Influence of the choice of the probabilistic model at the quantization stage.

data.

To make some comparison, we also show in Fig. 4.6 the performance of our method if the label images were encoded with the PNG format. It would be much less efficient. However, we must admit that at the present stage of development of the arithmetic coder, the encoding computation time of this arithmetic coder is not acceptable for practical applications (about 20 seconds for 5 codewords and a 800x400 image). On the other hand, The PNG compression scheme is much faster (less than 2 seconds).

We can also look at the performance of the coder when encoding label images quantized with the full probabilistic model. Fig. 4.7 shows that the coder does not

perform as well in this case. It seems then that the probabilistic model used in the vector quantization and in the arithmetic coder must match for optimal performance.

4.5.2 Visual performance

We have shown that our encoder yields excellent performance in terms of distortion–rate tradeoff. However, for the compression of images, we are far more interested in the visual performance, which is much harder to measure. In this respect, we achieved encouraging results.

A first very positive result is a noise reduction effect. In our test image "beauharnois." the seal color underwent distortion when scanned. Originally dark red, it shows on the image some light red areas. After quantization with our CECVQf algorithm, the seal recovers a homogenous color as shown in Fig. 4.8, column (C).

With respect to the luminance, the JPEG 2000 coder can encode it up to a compression ratio of 100:1 with small visual distortion. Beyond this value, the visual degradation is very important.

The chrominance is then proving to be a small part of the image information, which validates our first approach for image quantization though chrominance quantization. An encoding scheme that would deal efficiently with the chrominance part could compress it to a fraction of the luminance part.

4.6 Limitations and artifacts

Our encoding scheme suffers a number of bad visual effects even if it achieves excellent distortion–rate performance. The two main defaults are the non-detection of color areas and the spreading of some others.

4.6.1 Non-detection

What we call non-detection is the non-detection of a chrominance area in the quantization process that leads to an important visual change of this area between the

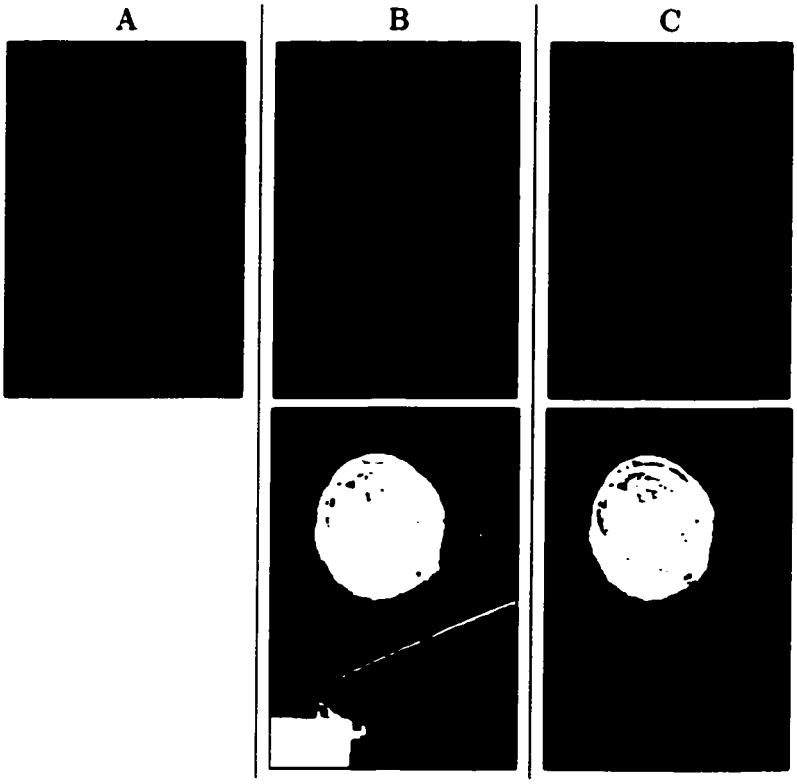

Figure 4.8: Quantization effects on the color seal for a compression ratio of 300:1 and different codebook size (from left to right: (A) original image, (B) image quantized with 3 codewords and the corresponding label image, (C) image quantized with 4 codewords and the corresponding label image)

original image and the quantized image (even when the luminance component is not compressed at all).

The first reason for non-detection to appear is the vector quantization of the image with too small a codebook. If the number of codewords is less than the number of easily identifiable color areas in the image, color areas will be merged leading to a wrong color for both areas. In practice, to avoid this problem, the size of the codebook must be slightly superior (by one or two) to the expected number of colors in order to detect all main areas, since large areas such as the background usually use in our scheme two or more codewords for the same color.

For example, in the "beauharnois" image, we can consider that there are only three main different colors (the background color, the seal color and the tape color) as we already discussed in Section 4.2.1. However, with less than four codewords, only two areas are detected (the tape area is merged with the seal area). This effect is shown in Fig. 4.8. Whereas with 4 codewords the seal color is enhanced, with 3 codewords, the seal undergoes noticeable color change.

The second reason for non-detection, are color areas, which are not associated with a codeword even if the codebook is large enough. This is particularly the case of the handwritten text color. Because this area is essentially thin our context-based method tends to include it in the background. Only the thicker parts are detected. The color of the text is then not as homogenous as the original and does not usually have its own color. Fig. 4.9 shows that handwriting is not well separated by the quantization process. This leads to color changes from one point to the other and can be a disturbing effect.

4.6.2 Overflowing

We call overflowing the spreading of a color area over another. This is mainly due to the context-based approach. If the weight of the entropy gain is big enough over the distortion measure, pixels might be associated with the wrong color area. This

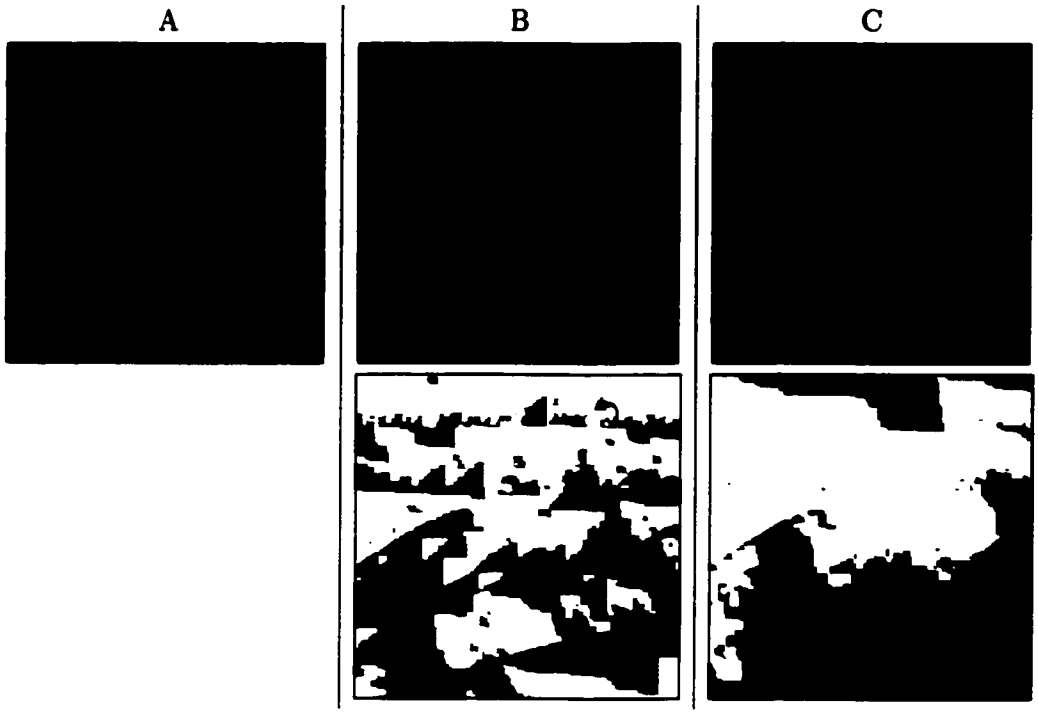

Figure 4.9: Non-detection of handwriting for 4 codewords (from left to right: (A) original image, (B) image with chrominance information compressed at 135:1 and the corresponding label image, (C) image with chrominance information compressed at 289:1 and the corresponding label image)

is usually done in the opposite direction of the neighborhood, but the use of a non-causal symmetric neighborhood does not seem to resolve the problem either, since overflowing could then appear in all directions. The spreading of a color over another leads to unpleasant artifacts, especially when they appear at the edges of physically different objects (such as tape on paper) that the eyes easily identify as different areas. An example of overflowing is shown in Fig. 4.10.

More than anything, this effect seems to be directly linked to our choice to make distortion–rate tradeoff a priority over the visual aspect. From these results we can expect that another method that would first properly identify the different areas by edge detection and only then associate a unique codeword for each of these areas, could attain homogenous areas suitable for context-based compression with good visual quality, even though the numerical distortion measure would be larger. This analysis should help to find different methods to handle the same problem, that is vector quantization of the chrominance information. But the segmentation should not be based on distortion but rather on the way the human eye naturally segments an image into different areas, based on the edges and on the colors.

4.7 Other axes of research

As said before in Section 4.6, vector quantization of chrominance based on minimization of distortion and entropy does not give good visual effects. We try here to suggest other schemes that we started to explore but need further investigations.

4.7.1 Fusion of chromatic components into a unique image

The chrominance information is a 2-dimensional space (CbCr space for instance). But each component is limited to a certain range ($[-0.5; 0.5]$ for the Cb and Cr components or $[0; 255]$ in software implementation). The 2-dimensional space can then be expanded in a 1-dimensional space, that is a scalar space in which numerous segmentation algorithms are available.

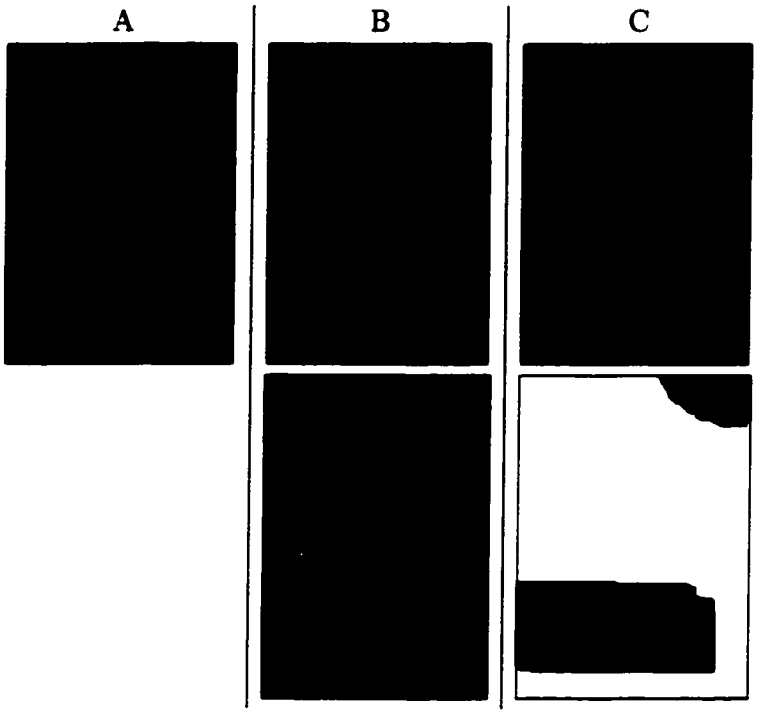

Figure 4.10: Spreading of a color area over another for 4 codewords (from left to right: (A) original image, (B) image with chrominance information compressed at 289:1 and the corresponding label image, (C) image with chrominance information compressed at 2785:1 and the corresponding label image)

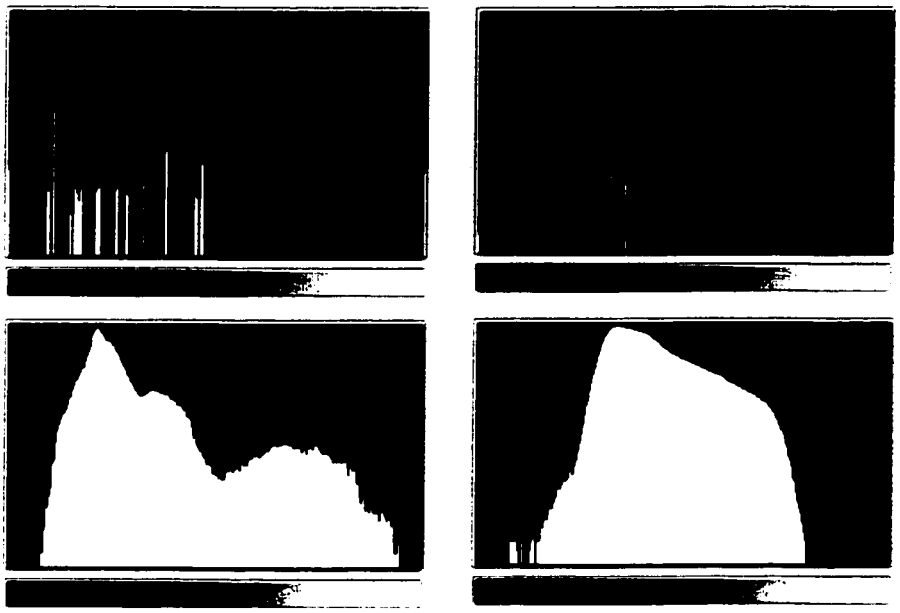

Figure 4.11: Histograms before and after slight blur for the beauharnois (right) and the december (left) images

If we consider that the two components of the chrominance space are integers in the range $[0; 2^n - 1]$, which is usually the case in software implementation with $n = 8$. the unfolding transformation $c = c_1 + 2^n * c_2$ is a reversible transformation. The reverse is simply the Euclidean division of c by 2^n. The remainder gives c_1 and the quotient gives c_2. The range of the unfolded space is $[0; (2^n - 1) + 2^n * (2^n - 1)]$, that is $[0; 2^{2n} - 1]$.

We can then consider the histogram of this 1-dimensional space. To have a continuous spectrum. we suggest to apply a slight gaussian filter, that will more easily reveal the shape of the spectrum as shown in Fig. 4.11. We notice that this spectrum seems to be the addition of several gaussian-like curves. It could be the addition of several distinct chrominance values to which gaussian-like noise was added. The initial chrominance values can be extrapolated from the top of the curves and usual deconvolution methods could be applied to determine to which chrominance areas each pixel belongs to.

4.7.2 Watershed segmentation

The watershed segmentation algorithm is a powerful tool widely used in medical imagery processing to isolate different elements in a medical image. It is used on 2-dimensional and 3-dimensional scalar (gray level) images with very good results to separate the different areas.

A new version of the algorithm could be developed to be applied to vector images but, even simpler, the existing algorithms could be applied on the scalar chrominance image obtained previously in Section 4.7.1. After the segmentation, the average value of each area becomes the codeword of this area. Over-segmentation can be reduced by the fusion of clusters under a certain criterion like distortion. The CECPNN algorithm discussed in Section 3.4.1 would be a good candidate for that.

With this method, we can expected to obtained a segmentation that faithfully conforms to what the human eye naturally does. The areas are made homogenous to achieve maximum efficiency when compressed with a context-based arithmetic coder, and spreading artifacts should not appear.

Chapter 5

Conclusion

5.1 Summary of the work

After studying the literature related to image coding, we found four existing image formats that could be of some interest for document coding.

JPEG 2000 Part 1 has been proving to be an excellent and very versatile image compression format, supporting numerous features of strong interest. It is not however by itself suitable for compression of typed or handwritten text.

Compound documents can then be encoded under the Mixed Raster Content (MRC) formats with a combination of JPEG 2000 for the background and the foreground and JBIG2 for the mask layer. DjVu and LDF implement this format with different segmentation algorithms and policies that result in important visual quality differences. MRC support is expected to be fully implemented in JPEG 2000 Part 6 in a couple of years. This technology is very promising but still needs some maturing to provide encoding with a visual quality matching the requirement of archive document researchers.

Then, we developed a codebook design algorithm based on the Binary Tree Partitioning algorithm and on the Conditional Entropy-Constrained Vector Quantization algorithm. The best implementation is pointed out through analysis of the distortion-rate performance of the algorithm. The Conditional Entropy-Constrained Pairwise

Nearest Neighbor (CECPNN) algorithm and the Conditional Entropy-Constrained Tree-Structured Vector Quantization (CECTSVQ) algorithm are two other promising algorithms based on the CECVQ algorithm that are worth being tested on images even if we did not have the time to do it.

Finally, we present our image compression scheme which includes, first, the choice of an adapted color space in which chrominance is not correlated to luminance and follows as much as possible a piecewise-constant model, second, vector quantization with the CECVQ algorithm, and finally, entropy coding. Even if we only encoded the chrominance information with our method it seems reasonable to think that a similar approach can be applied to the luminance component. The overall method proved to be interesting since contours and edges stay sharp as opposed to JPEG 2000 encoding. It has actually the same spirit as color reduction followed by PNG encoding, with better performance due to the constraint on entropy.

5.2 Contribution of the thesis

Even though we mainly used two images (the beauharnois and the december images shown in Appendix A), because these images are typical and representative of the type of document images we are dealing with, we believe that the conclusions and the evaluation results based on these two images are applicable to other archive images of this class.

We ran new tests on recent image formats for document delivery: JPEG 2000, DjVu and LDF. We pointed out that DjVu and LDF have interesting capabilities but only JPEG 2000 Part 6 is expected to bring a real solution.

We successfully adapted the Conditional Entropy-Constrained Vector Quantization (CECVQ) algorithm to image quantization and obtained excellent distortion–rate performance. We also concluded that the best tradeoff between distortion–rate efficiency and computing time is to run the modified algorithm (we called it CECVQm) only once for a given Lagrangian multiplier.

We proposed a new approach for compression of images with few colors. As far as it concerns its overall compression–rate performance, it shows very encouraging results. we also pointed out some visual artifacts due to the method which would need to be resolved.

5.3 Future research

We used the YCbCr color space but we believe that another color space, which respects more the piecewise-constant model, could give better results, especially for the recognition of the handwriting color. Also, despite the excellent performance of the CECVQ algorithm, we must point out that disturbing artifacts appear at low bit rate, when the constraint on conditional entropy is particularly strong. We believe the CECTSVQ algorithm could attenuate these effects by conserving more easily the intrinsic structures of the image. Other methods based on watershed segmentation or histogram deconvolution could also be considered, followed by the CECPNN algorithm to reduce oversegmentation.

Finally, a lot of work has still to be done on archive document images to find an image format completely adapted to their characteristics. Archive document images often suffer from bleed-through, that can even make reading difficult. However, it could also be exploited by conjointly coding the verso and the recto sides of the document. Another issue is multi-resolution coding. This capability is inherent to the nature of JPEG 2000 and a new format without this feature would stand little chance against it. Finally, since we are dealing with electronic format of archive documents owned by a certain institution, copyright protection can be a key issue. Adding visible and/or invisible watermarks resistant to attacks must be a possibility to consider in the case of these document images with low complexity.

Bibliography

[1] M.D. Adams and F. Kossentini. *JasPer: A software-based JPEG-2000 codec implementation*, Proc. IEEE Int. Conf. Image Processing (Vancouver. BC, Canada), October 2000.

[2] K.U. Barthel, S. McPartlin, and M. Thierschmann, *New Technology for Raster Document Image Compression*, Proc. SPIE (San Jose, CA, US), January 2000.

[3] R.E. Blahut, *Principes and partices of information theory*, Readings, Addison-Wesley, Massachusetts, 1987.

[4] L. Bottou, P. Haffner, P.G. Howard, P. Simard, Y. Bengio, and Y. LeCun, *High quality document image compression with DjVu*, Journal of Electronic Imaging **7** (1998), no. 3, 410–425.

[5] P.A. Chou and T. Lookabaugh. *Conditional Entropy-Constrained Vector Quantization of Linear Predictive Coefficients*, Proc. IEEE Int. Conf. Acoustics Speech Signal Processing (Alberquerque, NM), April 1990.

[6] P.A. Chou, T. Lookabaugh, and R.M. Gray, *Entropy-Constrained Vector Quantization*, IEEE Trans. Acoust. Speech Signal Process. **37** (1989), no. 1, 31–42.

[7] R.L. de Queiroz, *Compression of Compound Documents*, Proc. IEEE Int. Conf. Image Processing (Kobe, Japan), October 1999.

[8] R.L. de Queiroz, R. Buckley, and Xu M, *Mixed raster content (MRC) model for compound image compression*, Proc. SPIE Visual Communications and Image Process. (San Jose, CA, US), vol. 3653, February 1999, pp. 1106–1117.

[9] E. Dubois, J. Fadli, and D. Lauzon, *Vector quantization and coding of the chromatic information in an image*, Proc. SPIE Visual Communications and Image Process., vol. 3309, January 1998, pp. 832–841.

[10] E. Dubois, J. Konrad, and S. Cantet, *Estimation of Nonlinear Transfer Curves for Conversion of Color Images to a Known Color Space*, Proc. IEEE Int. Conf. Image Processing (Santa Barbara, CA, US), vol. 3, October 1997, pp. 26–29.

[11] T. Ebrahimi, C. Christopoulos, and D.T. Lee, *JPEG 2000*, Signal Process., Image Commun. **17** (2002), no. 1.

[12] A. Knoll, *New image formats and approaches for document delivery and their comparison with traditional methods*, 7th Conference on Professional Information Resources (Praha, Czech Republic), National Library of the Czech Republic, May 2001.

[13] D. Lauzon, *Optimisation débit-distortion d'un encodeur vidéo par représentation flexible du mouvement*, Ph.D. thesis, Université du Québec-INRS-Télécommunications, Octobre 2001.

[14] Y. Linde, A.Buzo, and R.M.Gray, *An algorithm for vector quantizer design*, IEEE Trans. Commun. **28** (1980), 84–95.

[15] M. Marcellin, M. Gormish, A. Bilgin, and M. Boliek, *An overview of JPEG-2000*, Data Compression Conference, March 2000, pp. 523–544.

[16] M.T. Orchard and C.A. Bouman, *Color Quantization of Images*, IEEE Trans. Signal Process. **39** (1991), no. 12, 2677–2690.

[17] W. Pearlman and D. de Garrido, *Conditional entropy-constrained vector quantization: high-rate theory and design algorithms*, IEEE Trans. Inf. Theory **41** (1995), no. 4, 901–916.

[18] W. Pearlman and L. Lu, *Conditional entropy constrained tree structured vector quantization with applications to sources with memory*, IEEE Trans. Inf. Theory. in review, 1998.

[19] A. Skodras. C. Christopoulos. and T. Ebrahimi. *The JPEG 2000 Still Image Compression Standard*, IEEE Signal Process. Mag. **18** (2001), no. 5, 36–58.

[20] ITU Recommendation T.44. *Mixed Raster Content (MRC) Mode*, 1997.

[21] W3C. *PNG (Portable Network Graphics) Specification*, http://www.w3.org/TR/REC-png, October 1996.

Appendix A

Test images

We provide here the original test images we used throughout this thesis to illustrate various effects. Both images were extracted from true archive document images provided by the The National Archives of Canada. These test images were chosen because they concentrate the main characteristics of most archive document images: paper texture, color handwriting, wax seal, library stamp, etc. They are the "beauharnois" image (Fig. A.1) and the "december" (Fig. A.2) image.

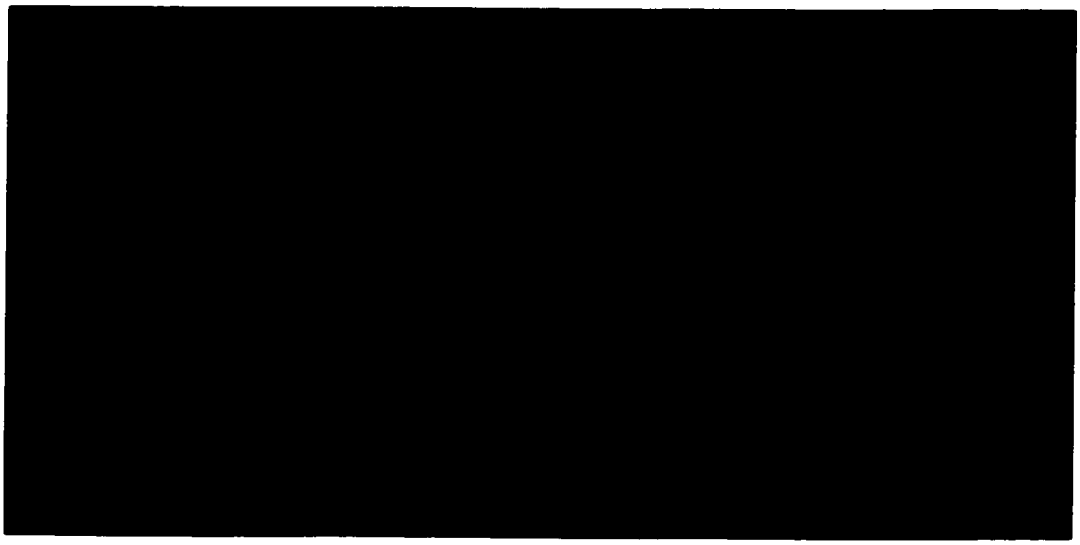

Figure A.1: The "beauharnois" image

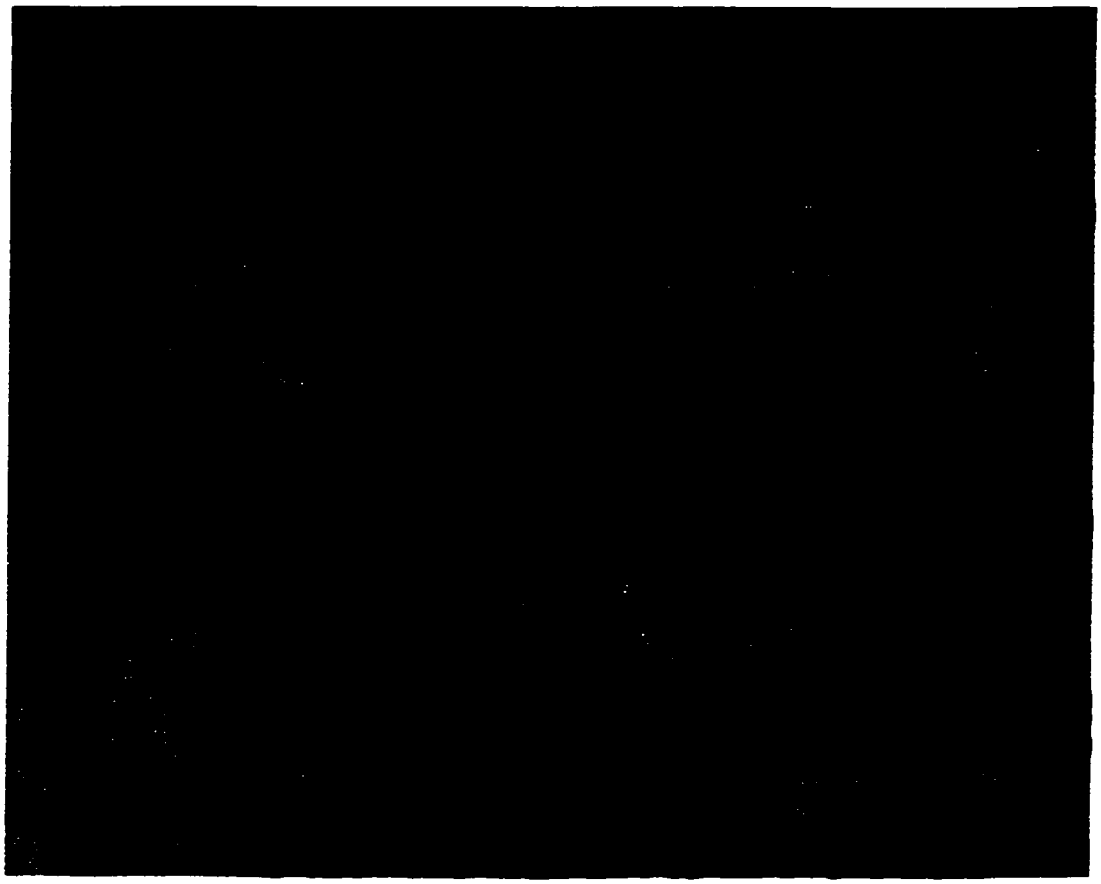

Figure A.2: The "december" image