



Università degli Studi di Cagliari

DOTTORATO DI RICERCA
IN INGEGNERIA ELETTRONICA ED INFORMATICA
Ciclo XXIII

**JPEG XR SCALABLE CODING FOR REMOTE IMAGE BROWSING
APPLICATIONS**

ING-INF/03 (Telecomunicazioni)

Presentata da: Bernardetta Saba

Coordinatore Dottorato Prof. Alessandro Giua

Tutor/Relatore Prof. Daniele Giusto

Esame finale anno accademico 2010 - 2011

*To Roberto
and my family*

Contents

1.	Introduction.....	1 -
2.	Image compression.....	4 -
2.1	The need to compress images.....	4 -
2.2	Compression techniques	5 -
2.3	Lossless compression.....	6 -
2.3.1	The Huffman algorithm.....	7 -
2.3.2	The RLE (Run Level Encode) algorithm.....	8 -
2.3.3	The LZW (Lempel-Ziv-Welch) algorithm.....	9 -
2.4	Lossy compression	9 -
2.5	Relative redundancy and compression ratio	10 -
2.6	Reversibility	10 -
2.7	Fidelity	11 -
2.8	The efficiency of the coefficient	11 -
3.	Color management	12 -
3.1.	The color	12 -
3.2.	The CIE colorimetric model.....	14 -
3.3.	Chromatic Adaptation.....	18 -
3.4.	The LAB color space	20 -
3.5.	RGB and CMYK color models	21 -
4.	State of the art	23 -
4.1	JPEG	23 -
4.1.1	How JPEG works.....	24 -
4.1.2	Decompression	27 -
4.1.3	Conclusion.....	28 -
4.2	JPEG 2000.....	29 -
4.2.1	How JPEG 2000 works	29 -
4.2.2	Introduction to the Discrete Wavelet Transform.....	30 -
4.2.3	JPEG 2000 in detail	31 -
4.2.4	The color transform.....	37 -
4.2.5	Strengths of the new algorithm.....	38 -
4.2.6	Comparison with JPEG	39 -
4.3	JPEG XR.....	40 -
4.3.1	JPEG XR algorithm	41 -
4.3.1.1.	Pre-scaling	41 -
4.3.1.2.	Color conversion.....	42 -
4.3.1.3.	Hierarchical image partitioning	42 -
4.3.1.4.	Structure of the Bit-Stream.....	43 -
4.3.1.5.	JPEG XR transform.....	44 -
4.3.1.6.	Quantization.....	49 -
4.3.1.6.1.	Quantization in the spatial dimension	50 -
4.3.1.6.2.	Quantization across frequency bands	50 -
4.3.1.6.3.	Quantization cross color planes	50 -
4.3.1.7.	Prediction.....	50 -
4.3.1.7.1.	DC prediction.....	51 -

4.3.1.7.2. LP prediction.....	51 -
4.3.1.7.3. HP prediction	52 -
4.3.1.8. Adaptive orders for coefficient scanning	53 -
4.3.1.9. Entropy coding.....	54 -
4.3.1.9.1. Adaptive VLC table.....	55 -
4.3.1.10. Decoder features	56 -
4.3.1.10.1. Sequential decoding	56 -
4.3.1.10.2. ROI decode	56 -
4.3.1.10.3. Spatial scalability	57 -
4.3.1.10.4. Quality scalability	57 -
5. Proposed Architecture	58 -
5.1 Client-server image browsing architecture	58 -
5.2 Error recovery algorithm.....	60 -
6. Experimental results	62 -
6.1 Region of interest (ROI) and tiling.....	62 -
6.2 Image scalability.....	65 -
6.3 Transmission over error channels.....	70 -
7. Conclusions.....	74 -
Bibliography	76 -
Related Papers	79 -

1. Introduction

Over the past 20 years, the industry that deals with digital images has received two major technological innovations that have profoundly changed the business world: the conversion to digital pictorial information and the Internet. The third innovation is playing recently: it is the mobile imaging.

The growth in sales of mobile terminals with integrated camera, has enabled hundreds of millions of people in a few years, to be equipped, always and everywhere, for a device for capturing digital images and their transmission network.

In addition, development in the field of digital memory makes it now possible to store on board hundreds if not thousands of image files. These are just some aspect that shows the importance of the standards that determine the formats for the interchange of digital media such as photographs. These are just some aspect that shows the importance of the standards that determine the formats for the interchange of digital media such as photographs.

The pictures are by far the type of digital data more present and exchanged over the Internet (just think of a phenomenon as Facebook that has established itself as the instrument par excellence of virtual social networking), thanks to the facility with which they may be acquired and exchanged by people.

The platform that underlies image processing is always subject to continue and unavoidable changes due to the speed with which new needs emerge. In recent years, the use of digital photography has grown considerably: just think how many images are downloaded via Internet or stored in digital cameras and mobile devices every day.

The ISO/IEC JPEG standard [1] has long been the reference format for compressing digital images, as well as “the format much more” used in the scenario of digital media. In fact, it can be shown that between multimedia digital files delivered over the Internet, the number of JPEG files is the highest. JPEG provided a decisive change in image compression and expanded especially with the birth of the WWW that needed formats for faster data transmission over the Internet.

However, today, this format is not able to support the continue evolution of acquisition attributes of existing digital media. In fact, the explosion of available Internet services, from e-commerce and

new digital devices, in particular by the spread of mobile phones and PDAs, tend to highlight the limits of the long established technology in the evolution of new scenarios.

The subsequent JPEG2000 [2][3] is currently gaining significant success mainly in digital cinema applications and in some areas of remote sensing and processing of medical images. It should be stressed however that the JPEG is very dated, as it dates back almost 15 years ago when both capture information and processing devices had performance very different from today

Despite being used in all cameras, both consumer and professional, the JPEG has a quality not at the state of the art and is unable to manage the dynamics of modern acquisition sensors, which allow for dynamic addition to 12 bpp per color against 8 bpp managed by JPEG.

This limitation does not allow to optimize the quality of data stored and/or to achieve optimal performance depending on the type of device (It should be stressed however that the JPEG is very dated, as it dates back almost 15 years ago when both capture information and processing devices had performance very different from today)..

The new coding algorithm JPEG-XR [4][5], can overcome various limitations of the first JPEG algorithm and can provide viable alternatives to the JPEG2000 algorithm. It has been developed for end-to-end digital imaging applications, to encode numeric data in a very varied range, with or without loss, with different color formats, reaching benefits that make it extremely interesting for the world mobile communications, in particular for internet browsing application of great images (for example scans of painting databases). It is designed to ensure image quality and compression efficiency, and to reduce complexity encoding and decoding operations.

This thesis is organized as follows. Chapter 2 presents a brief literature review of image compression algorithms. Chapter 3, briefly introduces the color management, that is a process where the color characteristics for every device in the imaging chain is known precisely and utilized to better control color reproduction . Chapter 4 provides an overview of image standard algorithm such as JPEG, JPEG 2000 and JPEG XR.

Chapter 5 presents a study of an interactive high resolution image viewing architecture for mobile devices based on JPEG XR. Display resolution, resolution scalability, image tiling are investigating in order to optimize the coding parameters with the objective to improve the user experience.

JPEG XR architecture is based on a reversible color space conversion, a reversible biorthogonal transform and an entropic non-arithmetic compression scheme. However, JPEG XR bit-stream is strongly vulnerable when transmitted over an error channel even when affected by very low error-rates. An analysis of channel errors effects to JPEG XR code-stream transmission is presented. Chapter 6 contains the experimental results. Region of interest specified by the user are progressively downloaded and displayed with an approach aiming to minimize the transferred

information between the request and the image presentation. Server side images are stored in frequency mode order and exploiting partitioning of images into tiles.

So, at first has been studied the image quality and has been compared with the other technologies available on the market (typically, those defined by the legislation JPEG and JPEG2000); was then evaluated the aspect relative to the efficiency of compression, verifying the overt less complexity in the operations of encoding and decoding. Experimental tests are performed on a set of large images and comparisons against accessing the images without parameter optimization are reported.

Lastly, the performance of the recently standardized JPEG XR in transmissions over channel affected by errors were analyzed.

Chapter 7 concludes this thesis.

2. Image compression

2.1 *The need to compress images*

The significant progress in many aspects of digital technology, especially within the area of image acquisition, data storage, printing and display, have led to the creation of a large number of multimedia applications relating to digital images, such as image browsing.

A big obstacle to the development of many of these applications is given by the large amount of memory required to directly represent a digital image (a digitized version of a single television frame contains in the order of one million bytes). Therefore, problems related to the high cost of storage and data transmission encountered.

The modern compression technologies offer a solution to the problems mentioned above by drastically reducing the amount of data that contain the information and providing very efficient tools for high image compression rates.

Image compression is the generic name under which the algorithms and techniques that are used to reduce the size of digital images are grouped.

An image is represented in a digital format as a series of dots (pixels) arranged like a chess board. Each point can use one or more bits to define the dot color. The compression techniques use the image peculiarity to reduce the local entropy of the file in order to make smaller the file.

The first compression techniques were very simple because the power of the first computers was limited and therefore they had to be simple to obtain acceptable decompression times. A simple but popular technique was the Run-Length Encoding. This provided the storage of a uniform color using special strings. For example, if there are thousand points of the same color in an image, the compression program saves first the color, then a special character and finally a number of points to paint with the same color; IFF images were saved with this method. Initially, the images had a limited number of colors, and so techniques that took advantage of this were used. Many formats used compression techniques based on symbols such as LZW. These algorithms built dictionaries containing groups of points that were frequently repeated and then memorized the image using the created dictionary. The GIF format uses this technique.

It should be noted that these compression techniques are all without loss of information.

Thanks to the growth of computing power, the computers have been able to handle images with thousands or millions of colors. Images with so many colors were badly managed by classical methods of compression, since the assumption of the small number of colors failed. So, new compression techniques, mainly to information loss, were developed. These new techniques allowed also compressions to 500% with the maintenance of an acceptable quality. Among the various methods the most popular became the JPEG.

The JPEG format uses various compression techniques combining them to obtain very high compressions. Among the techniques used, the most responsible for the compression (and loss of quality) is the discrete cosine transformed. The discrete cosine transform converts the screen points in their equivalent frequency domain representation. The resulting signal is formed by low frequency and high-frequency signals. The low-frequency signals represent the uniform color areas of the image while the high frequency components represent the details of the image and the quantization noise. JPEG compression will save the low frequency components and a part of the high frequency components. The file size grows proportionally with the increase of the quality and so with the saved components.

The success of the JPEG standard led many companies to develop solutions based on the compression loss, but with greater compression ratios than JPEG, at the same quality. These formats make use of very advanced mathematical techniques such as fractal compression, or those based on Wavelet transforms.

2.2 *Compression techniques*

The modern methods of image compression can remove unnecessary data without loss of quality; the information associated to them has one of the following two types of redundancy.

1. **Statistical redundancy**, related to phenomena such as correlation and periodicity of the data; it is also known as simply redundancy and can in turn be of two types:
spatial redundancy, when the value of an item can be predicted from the knowledge of the values of contiguous elements at the same instant of time;
temporal redundancy, when the value of an element can be predicted according to the values assumed by the same item in different time instants. This kind of redundancy may be removed without any loss of information.
2. **Redundancy subjective**, related to psycho-physical characteristics of auditory and visual systems; it is also known as irrelevance. It concerns the information area that can be

removed without obvious consequences in terms of perception. The removal of irrelevance is irreversible.

The compression techniques of digital data can be classified into two great categories: the first one aims to reduce the redundancy present on the signal and is used in lossless or non-destructive compression; the second one aims to reduce the irrelevance present on the signal and is used for lossy or destructive compression.

2.3 Lossless compression

Lossless compression techniques allow to regain a representation which is numerically equal to the original.

Since the quality of the reconstructed data is high, the compression factor reachable with these methods is not very high; generally ratios of 2:1 are obtained.

The most common lossless techniques are listened below:

- *Entropy coding*, which exploits the statistics about data. The most widely used is the Huffman code, with which it seeks to depict the most likely values with symbols of minimum length and to use symbols of greater length for values less frequent.
- *Run-Length encoding*, which aims to represent a sequences formed by the repetition of the same value with few bits. It represents the sequence with the pair (u, n), if a u value with n successive occurrences has been encountered.
- *Predictive Coding*, which is based on the high correlation of spatially or temporally adjacent data. It consists in predicting the value of a data to be encoded according to data already transmitted. If the prediction is well done the entropy of the prediction error is less than that of the original data and this results in a saving of bits in the coding phase.
- *Transform coding*, in which a linear transformation of data blocks is made, instead of considering data in their time and/or spatial original domain. This transformation redistributes the energy within data without altering them and trying to have the properties of correlation and repetition to allow a better performance of the methods first examined. Among the transformed most commonly used there is the discrete Fourier transform (Discrete Fourier Transform, DFT), discrete cosine transform (Discrete Cosine Transform, DCT) and the discrete wavelet transform (Discrete Wavelet Transform, DWT).

The Huffman algorithm

This non-destructive algorithm was invented in 1952 by the mathematician DA Huffman and it is a particularly ingenious compression method. It analyzes the number of occurrences of each constituent element of a file to be compressed: the individual characters in a text file and the pixels in an image file.

It unites the two elements less frequently found in the file in a sum-category which represents either. Thus, for example, if A occurs 8 times and B 7 times, it creates the AB sum-category, equipped with 15 recurrences. Meanwhile, the A and B components receive a different marker that identifies them as elements of an association.

The algorithm identifies the following two less frequent items in the file and puts them together in a new sum-category, using the same procedure described in step 2. The AB group can in turn enter into new associations and to constitute, for example, the CAB category. When it happens, the A and B components receive a new identifier that extends the code that will uniquely identify them in the compressed file that will be generated.

Is creates, for subsequent steps, a tree consisting of a series of binary branching, inside of which the rarer elements within the file appear with greater frequency and in subsequent, while more frequent elements rarely appear. According to the described mechanism, the rare elements within the uncompressed file are associated with a long identification code. The elements that are frequently repeated in the original file are also the least present in the association tree, so their identification code will be as short as possible.

The compressed file is generated by replacing each element of the original file with the relative produced code at the end of the association chain based on the frequency of that element in the source document.

The gain of space at the end of the compression is due to the fact that the elements that are repeated frequently are identified by a short code, which occupies less space than that occupied by the normal encoding. Conversely, the rare elements in the original file receive a long code in the compressed file; it may require, for each of them, an area considerably larger than that occupied in the uncompressed file.

The compressed coefficient produced by the Huffman algorithm is obtained by the algebraic sum of the space gained by short encoding of the most frequent elements and the space lost with the long encoding of the most rare elements.

So, this type of compression is more effective if the frequency differences of the elements that constitute the original file are greater, while if the element distribution is uniform, poor results are obtained.

The RLE (Run Level Encode) algorithm

In Run Length Encode compression algorithm, each repeated series of characters (or run) is coded using only two bytes: the first is used as a counter, and it is used to store the string length; the second contains the repetitive element which constitutes the string.

It is possible imagine to compress a graphic file containing a large background of a single uniform color. Whenever the sequential analysis of the file will encounter in strings of identical characters, the repetitive series can be reduced to only two characters: one that expresses the number of repetitions, the second the value that is repeated. The space saving will be directly proportional to the uniformity level present in the image.

If the RLE system was used on a photo full of different colors and soft transitions, the space saving will be much smaller, because the repetition strings that the algorithm will be able to find by sequentially reading the file, will be few.

Consider, finally, the limiting case of an image artificially created, like the one below, containing a set of pixels all different from each other in chromatic values. In this case, the use of RLE compression proves even counterproductive.



Figure 2.1 Enlarged image of a file of 16 x 16 pixels, consisting in 256 different unique colors.

This file, saved as an uncompressed BMP format, occupies 812 bytes; instead, if also the RLE algorithm is used, it occupies 1400 bytes, ie 1.7 times its original size.

The LZW (Lempel-Ziv-Welch) algorithm

The non-destructive algorithm that goes under the name of LZW is the result of the changes made by Terry Welch in 1984 to the two algorithms developed in 1977 and 1978 by Jacob Ziv and Abraham Lempel, and called LZ77 and LZ78, respectively.

The functioning of this method is very simple: it creates a dictionary of strings of the recurring symbols in the file, constructed such that at each new term added to the dictionary is coupled a single string in an exclusive manner. There is a departure dictionary consisting of 256 symbols of the ASCII code, which is increased by the addition of all the recurring strings in a file, that are greater than one character. A short code is stored in the compressed file and it unequivocally represents the string entered in the dictionary thus created.

There is, of course, a set of not ambiguous rules for the dictionary encoding , which will allow to decompression system to generate a dictionary exactly equal to that of departure, so as to be able to perform the inverse operation to that of compression, consisting in the replacement of compressed code with the original string. The complete and accurate reversibility of the operation is essential in order to regain the exact content of the original file.

The space saving in a compressed file with LZW depends on the fact that the number of bits needed to encode the "word" that represents a string in the dictionary is smaller than the number of bits needed to write in the uncompressed file all the characters that compose the string. The compression coefficient of the file will grow in a proportional manner to the length of the string that it is possible to insert in the dictionary.

The most popular graphics formats that use the LZW algorithm are TIFF and GIF.

2.4 Lossy compression

The use of lossy techniques implies a certain approximation of the original data.

The resulting quality loss is rewarded by high compression ratios, from 20-200:1 even up to 1000:1. For this reason, these techniques are preferred to those without loss.

In particular, it is evident the need to use the lossy formats (precisely in relation to these their higher compression capabilities) for displaying images on the Internet, subjected to the slowness of the modem connection.

The most important lossy techniques is certain the quantization. It may look like a surjective but not injective function that is applied to data. It is accomplished by dividing the domain into a certain number of subsets and assigning a quantization value to each one. It is called scalar or vector quantization, depending on the size of the domain.

2.5 Relative redundancy and compression ratio

The data redundancy (subjective or statistical) is an essential element for compression. This is not an abstract concept, but a mathematically quantifiable entity.

In fact, named n_1 and n_2 the number of bits needed to represent the same information in two different data sets, the relative redundancy of the first data set compared to the second is defined as:

$$R_D = 1 - \frac{1}{C_R} \quad \text{where CR is the compression ratio, defined as} \quad C_R = \frac{n_1}{n_2}$$

If $n_1 = n_2$, $C_R = 1$: the first representation of information does not contain redundant data (compared to the second).

If $n_2 \ll n_1$, $C_R \rightarrow \infty$ and $R_D \rightarrow 1$: the first representation is highly redundant, the compression ratio can take significant values.

If $n_2 \gg n_1$, $C_R \rightarrow 0$ and $R_D \rightarrow -\infty$: the second representation contains much more data than the first; it is an expansion rather than a compression of data.

In general, it therefore has : $0 < C_R < \infty$ and $-\infty < R_D < 1$. In practice, a compression ratio equal to 10 means that the redundancy is 0.9, so the 90% of the data of the first performance can be removed.

2.6 Reversibility

Now consider a typical graphic file, such as a color photograph saved in RGB mode. This mode requires the use of three bytes to store the color and brightness information for each point, or pixels, of the image. It is simply necessary multiply by three the total number of pixels that constituted the image, to find out the space on disk occupied by this graphic file. This number is obtained, in turn, multiplying the number of columns for the number of rows that contain the rectangle occupied by the image.

A photo of 1024 horizontal by 768 vertical pixels, occupies a space of $1024 \times 768 \times 3 = 2,359,296$ bytes, which is equal to 2,304 Kilobyte or, again, to 2.25 megabytes. It is important to specify that this figure represents the space occupied by the graphic file in uncompressed form.

Any program for monitor display of image will need all those bytes to generate on a computer screen a photo like those original stored on disk; they must also be prepared in the same exact sequence in which were stored the first time.

So, it is evident that a byte sequence that composes a compressed graphic file will not be used to generate the image contained in the original file.

This brings us to define a fundamental characteristic of any compression format liable to practical applications: the reversibility of the compression. Generally, if a program has the ability to save a file in a compressed format, it is also able to read files that were compressed with that particular format, resetting the information contained into them, ie decompressing.

Without reversibility, even the best compression algorithms is useless.

2.7 Fidelity

The main difference that we can establish between the compression formats of graphic images is given by the extent of their reversibility. A format that is able to return at the end of decompression, an image exactly equal (pixel by pixel) to the original is usually lossless. Conversely, a compression format that can not ensure absolute reversibility is defined as lossy. The thing that is lost or not is the fidelity of the restored image to the original one.

The professional graphics must know perfectly the characteristics of the compression formats that use, if they want to get the best from manipulations on files. It would be a serious mistake to save and re-save a file such as JPG in a lossy format, and then use it to a lossless format as the TIF. Instead, the opposite is correct, ie it is possible to save a file many times in a non-destructive format, and then save it only at the end in a destructive format, if necessary. The storage in a lossy format must always be the final piece of the transformation chain to which a file is submitted.

2.8 The efficiency of the coefficient

At this point an obvious question arises: why use a destructive compression format, if there are non-destructive systems that allow you to compress and decompress the same file over and over again, retaining all the information contained in it?

The answer is that in many cases the amount of space that a non-destructive (lossless) compression is able to save is much lower than the space savings achieved by a destructive (lossy) compression. The compression efficiency is calculated by dividing the original size file for those compressed.

Because of the different type of operations performed on the image, in certain cases it is appropriate to use a compression format destructive, in others a non-destructive. Typically, graphic images containing drawings with uniform tints and sharp and well contrasted edges are the ideal candidates for the lossless compression. In contrast, the photos, especially those containing a large number of different colors and soft transitions between foreground and background, require lossy compressions if you want to achieve a significant saving of space.

3. Color management

3.1. The color

First, it is necessary to introduce the color concept: *the color is a visual perception caused by a physical stimulus, the light, which directly or indirectly, affects the eye, (particularly the retina), arrive through the optic nerve arrive to the brain, where the sensation of color is created.*

The light is composed of monochromatic electromagnetic radiation of different wavelengths included between 400 nm (ultraviolet) and 700 nm (infrared). Every single monochromatic radiation is perceived by the eye as a single color, as shown in the figure below.

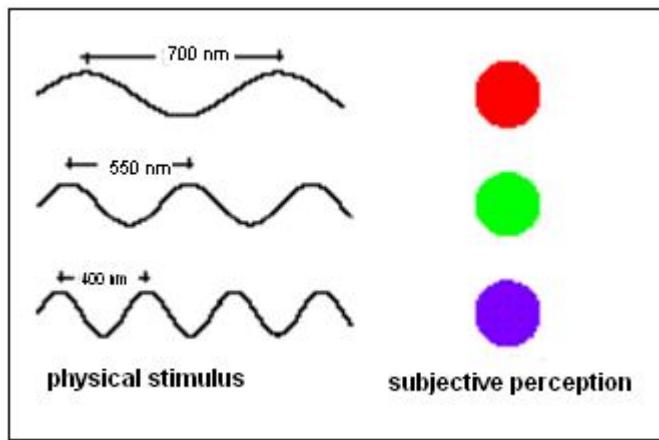


Figure 3.1 The eye perceives each monochromatic radiation as a color.

It is possible to build the color spectrum shown in Figure 3.2, putting below all visible monochromatic radiations with their wavelength, and the respective perceived colors:

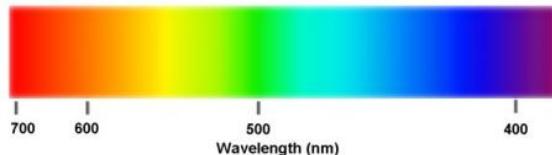


Figure 3.2 The spectrum of colors.

In this spectrum, however, some color like white, black, gray and many other are missing. The reason of this fact is that, normally, in nature, the lights are a mixture of different monochromatic radiations which are perceived by the eye do not individually but as a "total" color. These colors are called "non-spectral" and the white of sunlight, which contains all visible wavelengths, is one example.

It is important, therefore, to distinguish between two different worlds: one physical, objectively describable and measurable with precision and another one perceptual where subjective variables come into play.

Normally the way in which we see a color is influenced by several factors, which can be grouped into three main categories: psychological / emotional, physiological and environmental.

The psychological / emotional affects are related to how a person interprets a certain color. The blue and green colors, for example, are cold and have a relaxing effect; black and white are associated with the death and purity concept respectively.

These factors are very important in all those activities defined as "color-critical", like, for example, the advertising that tries to influence people taking advantage, too, the feelings aroused by the colors used in ads or products.

The physiological factors concern, however, the manner in which the receptors present in our eyes (cones) produce signals that determine the sensations of colors. The behavior of these "sensors" varies from individual to individual, and sometimes malfunctions that lead to the inability to distinguish certain colors can exist.

The environmental factors are related to light and colors of the place in which a color is seen. The figure below shows how the color perception can be strongly affected by the type of ambient light (on the left) and by the colors around it (on the right). This phenomenon is called "Achromatic Simultaneous Contrast": the red dots, although they are identical, seem to have different brightness.

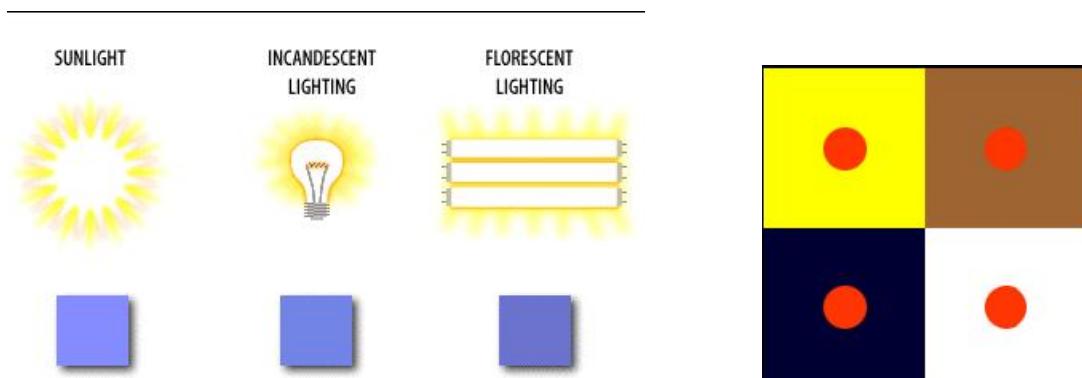


Figure 3.3 The perception of color is influenced by both ambient light and surrounding colors.

At this point, it is questionable what laws govern the color perception. The first to give an answer to this question was Isaac Newton (1642-1727), who in 1666 drew the diagram shown below, which he regarded as approximate.

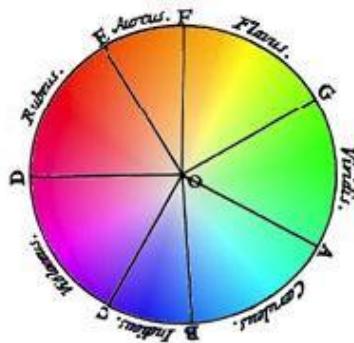


Figure 3.4 The Newton diagram.

Newton has decided to combine the extremes of the color spectrum, resulting in a circle. The spectral colors, in this model, are on the circle, while those not spectral are inside. The white is placed at the centre. A third dimension that goes perpendicular to the plane is implied: the brightness.

3.2. The CIE colorimetric model

The CIE colorimetric model, proposed by *Commission Internationale de l'Eclairage* in 1931, represents all and only the colors that the human eye can see and it is illustrated by the following figure.

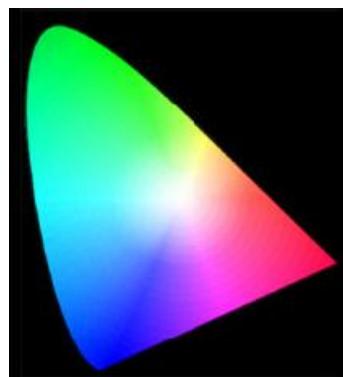


Figure 3.5 The CIE 1931 chromaticity diagram.

This chromaticity diagram, which is that of Newton, modified, updated and standardized, is two-dimensional: the white is in the centre and the saturated color of the light spectrum are along the

curve portion of the perimeter: in clockwise direction, red, yellow, green, blue, purple. The central colors are unsaturated (white is the most unsaturated) and peripheral colors are saturated. The diagram represents the colors (around the perimeter) and saturation (from the perimeter toward the centre).

Each color is represented by a point inside the horseshoe area. The entire area is included in a system of x, y Cartesian coordinates as shown below.

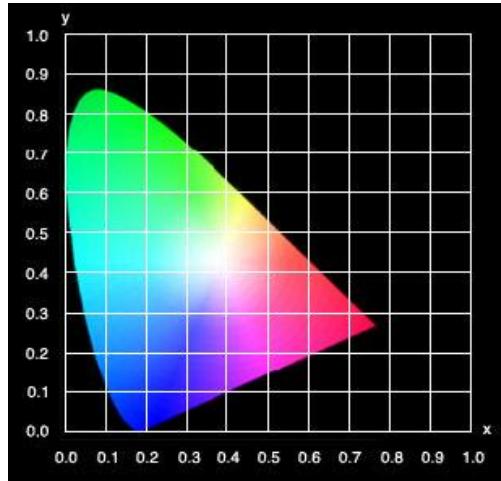


Figure 3.6 x, y coordinates in the CIE 1931 color space.

Both x and y coordinates take values from 0 to 1. At each color corresponds a pair of x, y coordinates, but not at all of the pairs of coordinates in the range corresponds a color. For example, there is a certain red defined by the coordinates $x = 0.6$ and $y = 0.3$, the color of (0.4,0.2) coordinates is a lilac while the (0.1, 0.6) color is a green.

Note that in the diagram the two x and y coordinates are denoted by lowercase, since in colorimetry X and Y case have another meaning.

Up to this point the brightness of the colors is not considered; it can be introduced by adding a third dimension to the just seen diagram.

In fact, the CIE 1931 chromaticity diagram is only a "slice" of a more complete space, the CIE 1931 color space to which the coordinates XYZ are assigned. This space has the form shown in the image below, where the chromaticity diagram is individuated (in a form slightly different from the one considered here).

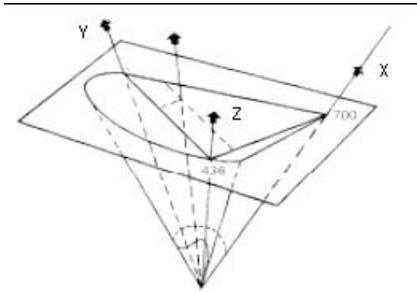


Figure 3.7 The XYZ space. Its projection on a plane is an xy chromaticity diagram.

The Y coordinate of the XYZ space, for how this space is constructed, expresses the brightness of considered color.

The relations to switch from three-dimensional XYZ coordinates to xy coordinates are:

$$x = X / (X + Y + Z), \quad y = Y / (X + Y + Z)$$

It is also possible to consider another coordinate system: the space xyY, obtained by adding the third dimension to the xy diagram, ie, the Y coordinate.

The two systems x,y,Y and X,Y,Z constitute two different coordinate systems for the CIE 1931 chromaticity diagram; they are mathematically linked to each other, and if you know the coordinates of a color given by x,y,Y system, it is possible to know her coordinates given by X,Y,Z system and vice versa.

The formulas to switch from xyY to XYZ, with a Y fixed are:

$$X = xY / y, \quad Z = (1-x-y) Y / y$$

The spaces x,y,Y and X,Y,Z allow, therefore, to express the same colors in the same CIE 1931 chromaticity diagram with two different but connected coordinate systems. You choose one of the two systems according of the use and convenience of data processing.

Before introducing an important property of the CIE 1931 diagram, we see at a glance how colors are produced with the additive mixture.

There are at least 3 types of additive mixture:

- Space;
- temporal average;
- spatial average.

The additive mixture space is obtained when more lights, which produce different sensations regardless of color, are superimposed. In this case, they produce a new color sensation. For example, if two lights that, which produce the sensation of red and green independently, are overlapped, the produced sensation will be of yellow (Fig. 3.8).

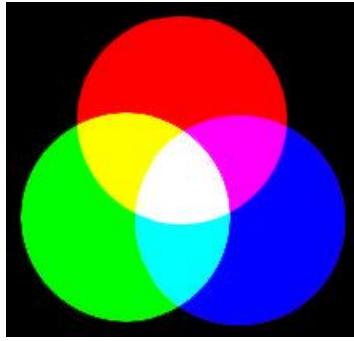


Figure 3.8 The additive mixture Space.

With the additive mixture in the time average, the different radiations strike the eye at different times, but so close together (at least 50, 60 times per second) that the eye draws a single media sensation.

For example, if a wheel with red and green colored areas is spun rapidly, a sensation of a brown color (dark yellow) will be produced..

The additive mixture in the spatial averaging is the most common case in practice. For example, monitors and televisions are based on additive synthesis, because the dots of R, G and B phosphors are neighbors and are so small as to be indistinguishable; in the eye and blend into a single point on the retina.

An important property of the CIE 1931 system is illustrated. If you mix two colors in additive synthesis, a third color is obtained; a series of other colors can be obtained by varying the proportions of the first two colors.

For example, starting with a green and a red (right below) and mixing them in different proportions various colors including yellow are obtained. How these colors are arranged on the CIE 1931 chromaticity diagram? Newton concluded that the different resulting colors are on a straight segment that joins the two initial colors, and this result is also valid in the CIE 1931 (left below).

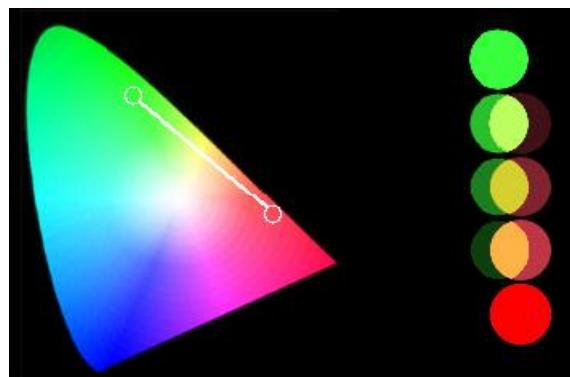


Figure 3.9 The additive synthesis of two colors on the CIE 1931.

This is an important property of the CIE 1931 chromaticity diagram: additively mixing two colors, represented by two points on the diagram, it is possible to get all colors that are on the segment connecting the two points, by varying the proportions of the two initial colors.

In particular, fixed two colors at certain luminosity, the brightness in their additive mixtures are added and the resulting color is in the barycentre of the brightness. If the brightness are the same, the color is in the centre.

So, if you mix a green and a red both with a brightness equal to 1, a yellow with brightness equal to 2, will get and this yellow will be half way between green and red.

By additively mixing three colors, represented by three points on the diagram, all the colors that are inside the triangle of which three points are the vertices, will get.

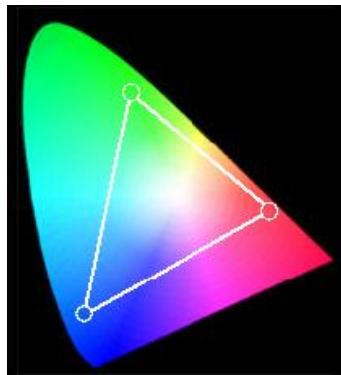


Figure 3.10 The synthesis of three additive colors.

The majority of the colors, even if not all, can be generated placing the three vertices equally spaced on the perimeter. This is the *trichrome principle*.

3.3. Chromatic Adaptation

The human visual system optimizes its response to the particular viewing conditions with a dynamic mechanism called adaptation; it can be of two types: color and luminance.

The adaptation to the luminance consists in a variation of the visual sensitivity when the light level increases or decreases. To realize this behavior, just pass by a very bright in a dark room: you can not initially see anything, then the eye begins to gradually adapt to darkness and see better.

The chromatic adaptation occurs, however, when an object under different lighting types (eg natural light first, then a light bulb) is observed: if the observer has had sufficient time to allow the eyes to adjust to new light source, the object retains its appearance under both illuminations.

The chromatic adaptation is, therefore, the ability of our visual system to discard the illumination color and to preserve in an approximate manner the appearance of an object.

Now consider a piece of paper and two light sources represented by the following coordinates in the xy chromaticity diagram: D65 (0.3127,0.3290) and D50 (0.3457,0.3585) (these are two of the standard light sources of the "D" series, introduced in 1963 by CIE. The standardization of the light sources has been carried out to allow to define the artificial light sources used in the industry to evaluate the colors).

- *Scenario 1*: the piece of paper illuminated by D65, has by definition the chromaticity coordinates of D65. An eye adapted to D65 sees by definition this white paper.
- *Scenario 2*: if the same piece of paper is illuminated with D50 its chromatic coordinates will become those of D50. Thus, for a measuring instrument, the piece of paper has changed color. But to the eye adapted to D50, the paper will appear as white as before.
- *Scenario 3*: if the piece of paper is modified so that the instrument sees it the same as before, the adapted eye will see a different color.

This is because measurement systems that rely on colorimetry does not take into account (or do not have the) the ability to adapt to the enlightening of the human eye.

The various tools to capture and display images must, therefore, deal with different types of light sources, since these can change switching from one device to a different one. A monitor, for example, may have a white point lies between the D50 and D93 coordinates, and, therefore, two different displays may represent the white with different coordinates in the xy chromaticity diagram (ie they have two different points of the white).

To create the appearance of a color image, all these systems have to apply a transformation to the input color (captured with some enlightening) to convert it in the output color .

And this is the purpose of transformations for chromatic adaptation. Applying these transformations to the X 'Y' Z values of a color under a given illuminant, we can calculate a new set of three X "Y" Z "coordinates related to a different illuminant.

All transformations use the same basic methodology:

1. transform the XYZ coordinates in the tristimulus values of the cone responses;
2. change those responses;
3. reprocess these values in XYZ.

The most commonly used transformations are:

- XYZ linear scaling;
- von Kries transformation (Johannes von Kries, 1878);

- Bradford transformation (Lam and Rigg, Ph.D. thesis, University of Bradford, 1985).

3.4. The LAB color space

Another color space that, such as the XYZ space, represents all the colors that the human eye can see, is the CIELAB space.

As shown in Figure 3.11, the LAB model is spherical and represents the colors through three components: the L axis is the brightness that varies from 0 (black) to 100 (white); the a and b axes respectively define the ranges of colors from red to green and from yellow to blue.

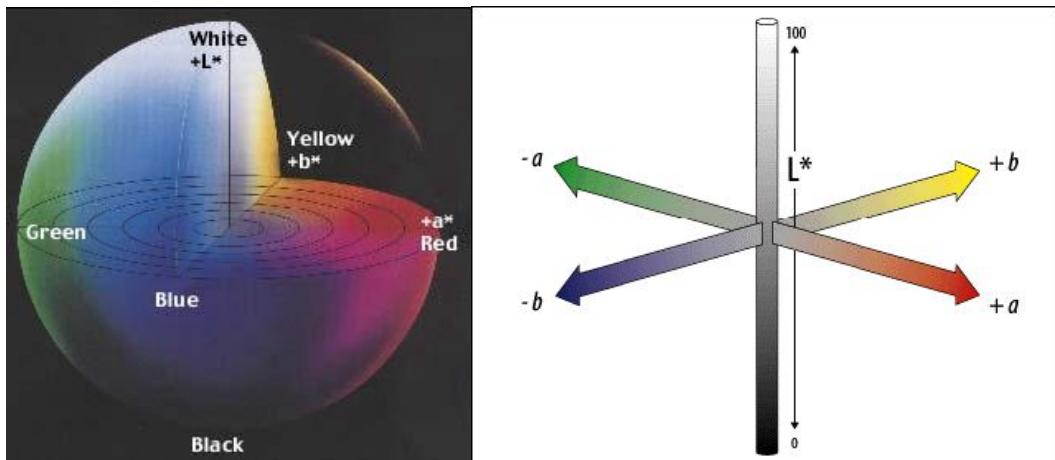


Figure 3.11 The CIELAB color space.

Note how the brightness is "separate from the color"; it means that it does not depend on the values assumed from the a and b channels. Instead, as we shall see, in the RGB and CMY spaces, the brightness is related to the values of the three channels.

It is possible to switch between the different XYZ, and LAB xyY coordinate systems using appropriate mathematical formulas, without loss of information.

3.5. RGB and CMYK color models

The RGB color space is the natural "language" of color description used by electronic devices such as monitors, scanners and digital cameras, that reproduce colors by transmitting or absorbing light rather than reflect it. The colors observed on computer monitors, for example, appear when the electron beams strike the red, green and blue phosphors of the screen causing the emission of different combinations of light.

The RGB color space is called additive because colors are generated by adding colored light to other colored light. The secondary colors are brighter than red, green and blue primary colors that are used to create them. In RGB mode the maximum intensity of the primary colors produces the white. A combination of identical amounts of red, green and blue produces neutral gray tones; dark grey tones are generated using low values of dark gray, while lighter tones are created with higher values. The figure below left shows the RGB space.

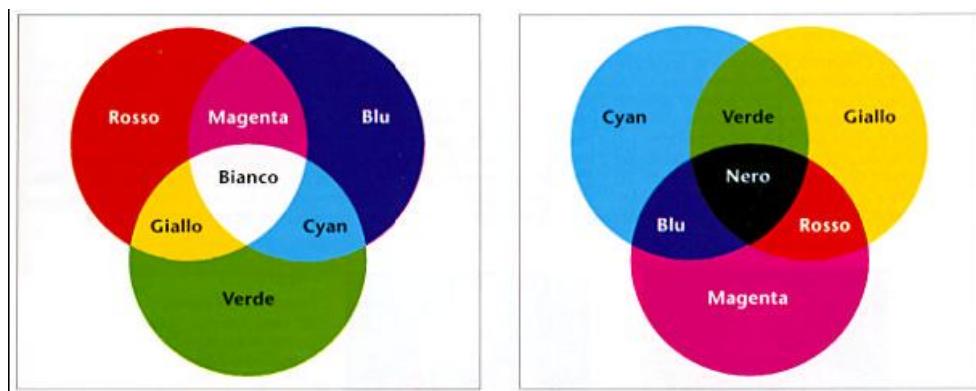


Figure 3.12 Color models RGB and CMY.

The right part of Figure 3.12, instead, represents the color space complementary to the RGB, obtained by subtracting the primary R, G and B colors from white light. This type of color representation is called CMY mode and it is the one used by the printing industry and, in general, by the printing devices. As can be clearly seen in the figure, the CMY space is subtractive: while in the RGB space the light is added to another light resulting in brighter colors, in CMY mode the light is subtracted, producing darker colors.

For completeness, it should be stressed that the RGB and CMY spaces are complementary to each other only in theoretical line. By combining, in fact, equal amounts of cyan, magenta and yellow neutral grays should form and combining the maximum quantity you should get the black (the opposite of white in the RGB model). This is true on monitors, but, in print, a muddly brown is obtained instead of the black and this phenomenon is caused by impurities in the pigments of the

inks. For this reason it is necessary to add another color, the black (denoted K by key color) that allows for deeper blacks and shadows with definite sharper. Obviously, the addition of a fourth color unbalances the equation for the direct conversion from RGB to CMYK making more complicated the correspondence between these two color spaces.

One last thing to remember is that the RGB and CMYK color spaces are device-dependent and, therefore, the colors generated by a device may differ from those played by another. All this does not happen, however, with the colors identified by CIE XYZ and LAB models that, such as visual perception, are not tied to specific devices and are much more numerous than those represented in RGB and CMYK :this is the reason for whicht he programs for the digital color management exploit the CIE as a base to convert the colors in sure way, then considering the limitations of the range of input and final output devices.

The diagram below shows the relationship between the viewed color mode.

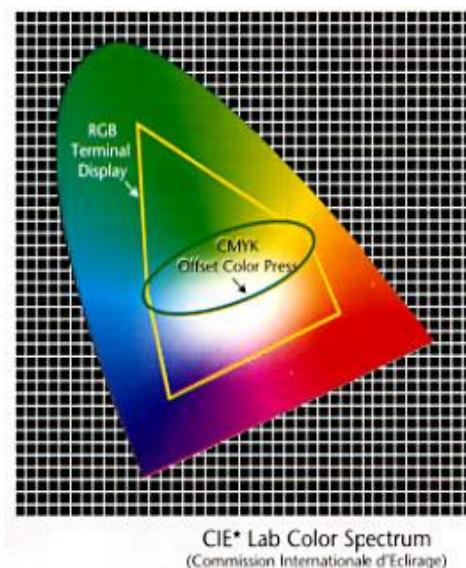


Figure 3.13 Relations between the CIE color spaces, RGB and CMYK.

4. State of the art

4.1 JPEG

In the late '80s the engineers of the ISO / ITU decided to form an international research team to create a new standard format of photographs recordings which allowed a strong reduction of the space occupied in memory at the expense of a more or less marked but always acceptable quality degradation. Thus the Joint Picture Expert Group was born and in 1992 released the popular JPEG compression format, which takes its name from this group.

Before the JPEG advent, BMP, TIFF, GIF, TARGA, the IFF and the other format images were typically used. All these formats, however, were marked by the loss-less compression, that is, without any loss of information and therefore were unable to obtain satisfactory compression ratios. For example, the BMP not apply any compression and simply provides an array of 'words', each of which defines the color of the corresponding pixel (eg using 8,16 or 24 bit); the TIFF instead apply entropic techniques such as RLE or Huffman compression (or LZW) to reduce the image size. These techniques have some effectiveness especially in those cases where not all the image surface contains relevant information; this occurs for example in images with large areas of a single color but it is difficult to obtain satisfactory compressions with photographic images. Finally, the GIF format was not suited to handle photographic images because it is limited to the use of a maximum of 256 colors. All of these methods, typically, did not allow to reduce the size more than a 2 factor. The JPEG was a decisive forward step in image compression, and quickly established itself especially with the birth of the WWW that necessarily needed a very simple format for transmission over the Internet. Note that an image of 500 x 500 pixels in full-color occupies something like 750KB in uncompressed format, while JPEG allows to reduce its size by a 20-30 (or even more) factor, depending on the level of chosen detail, that is the number of details discarded during the compression phase. The utility is obvious: the original image is changed from 750KB to only 30KB; it is then easily storable on magnetic supports and transferable over the Internet even using an ordinary modem.

How JPEG works

JPEG is a flexible standard that defines a set of possible processes to be performed on images, processes that can also be skipped. A guideline for compression and especially a rigid specifications for decompression is given. In practice, it is not specified how to do compression, but only the rules that must be observed by the compressed data, in order to obtain a correct decompression.

In any case there is a more or less standard process, which is illustrated in this image:

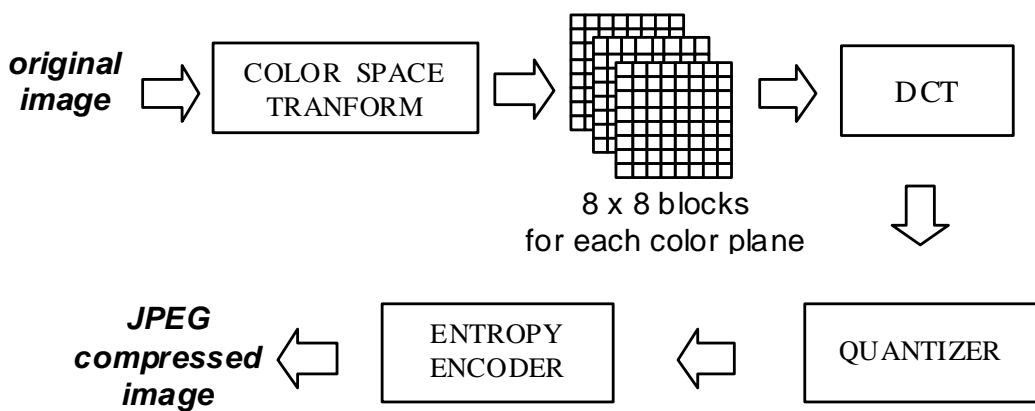


Figure 4.1JPEG encoder architecture.

a) Conversion of color space (optional)

The original image is converted from RGB to YIQ (or YUV) color space. The YUV format consists of 3 color planes: the luminance (Y) and two chrominance components (U and V). This separation, though not strictly necessary, allows for better compression. In fact, exploiting a phenomenon known in the video, you can reduce the size of the YUV image with a slight reduction in quality by applying a downsampling (decimation) of the chromatic components and keeping intact the luminance information.

The reduction is typically done by making the average, two by two, between adjacent pixels of the U and V planes; in practice the horizontal resolution of these planes is halved. In theory it is possible also to halve the vertical resolution (eg in MPEG1) but it is not a practice typically used in JPEG. In the first case we speak of 4:2:2 encoding and of 4:1:1 in the second case (at each block of 2x2 pixels in tY correspond 2x1 blocks in the first case and of a single pixel in the second case). Already this lossy arrangement (ie, with loss of information) allows a reduction of 30% and 50% respectively.

If this step is skipped, the next phase will process the image in RGB instead of YUV.

b) Frequency Analysis (DCT)

The key development of JPEG is certainly the DCT (Discrete Cosine Transform) in its two-dimensional (2D) version. The DCT is a transform which in general converts the signal from the time to the frequency domain. This is a version in the real range of the FFT which is instead in the complex domain. The resulting coefficients represent the amplitudes of those harmonic signals (cosine) that added together reconstruct the signal.

JPEG uses the two-dimensional version of the DCT and in this case there is no mention of time and frequency but only of space and spatial frequencies. In order to be processed, the image is divided into color planes (three color planes RGB or YUV depending on the stage before) and within each plane it is again divided into blocks of 8x8 pixels.

The block of 8x8 pixels in the domain of the 'space' is transformed into a block of 8x8 coefficients in the spatial frequency domain. In this block, the coefficients in the upper left represent the low spatial frequencies while those gradually in the lower right, represent the high spatial frequencies, that are the image details. In particular, the first coefficient of the transformed block represents the average of the values of the original 8x8 block (also called DC component).

The mathematical formula of 2D DCT is shown below:

$$B(k_1, k_2) = \sum_{i=0}^{N_1-1} \sum_{j=0}^{N_2-1} 4 \cdot A(i, j) \cdot \cos\left[\frac{\pi \cdot k_1}{2 \cdot N_1} \cdot (2 \cdot i + 1)\right] \cdot \left[\frac{\pi \cdot k_2}{2 \cdot N_2} \cdot (2 \cdot j + 1)\right]$$

The original image is large $N_1 \times N_2$ pixels; $A(i, j)$ is the intensity of the pixel at i, j position; $B(k_1, k_2)$ is the resulting coefficient. Typically, A is a number of 8 bits, and in output will have variable values between -1024 and 1023.

c) Quantization

The deletion of less important visual information occurs in this phase. This is accomplished by multiplying an 8x8 coefficients matrix in the frequency domain for a 'quantization table'. The table

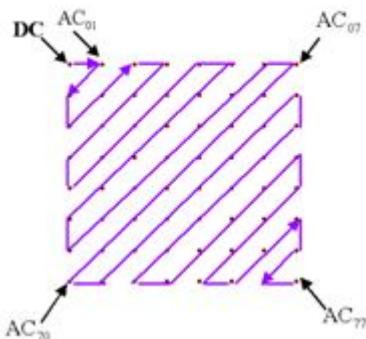
contains values between 0 and 1, the lowest are located in correspondence of the high frequencies while the higher ones in correspondence of the low frequencies.

The values thus obtained are rounded to the nearest integer; in this way the least significant coefficients tend to zero while the coefficients related to the information contributions most important remain. High-frequency values are often rounded to 0, since they are already small. The result is the concentration of a few coefficients different from 0 at the top left and 0 all the others.

When in a JPEG file you choose the compression factor, indeed you choose a scale factor on the values of 'quantization table'. The number of coefficients that are set to zero grows in proportion to the decrease of their values, with a consequent reduction in the number of significant coefficients. This process, of course, erases information more and more important and leads to a progressive deterioration of the compressed image quality.

d) Entropy coding

Once removed the less important details thanks to the DCT and quantization, it is necessary to adopt a series of entropic techniques to reduce the amount of memory required to transmit the remaining significant information.



It is important to separate the continuous component (indicated as DC in the figure) from the variable component (indicated with AC), among the remaining coefficients. The two types of values will be treated separately.

Figure 4.2 Zig-Zag sequence.

d1) Zig-Zag reading

The AC component is scanned by a Zig-Zag reading. Looking at the figure above it is clear what it means. The Zig Zag reading makes as much as possible adjacent the coefficients equal to 0 and allows an optimal data representation using Run Length Encoding.

d2) RLE on the variable component (AC)

Run Length Encode (RLE) is a simple compression technique applied to the AC components: the 1x64 vector resulting from Zig-Zag reading contains a lot of zero in sequence. For this reason the vector is represented by pairs (*skip*, *value*), where *skip* is the number of values equal to zero and *value* is the subsequent value different to zero. The pair (0.0) is considered as a signal of end sequence.

d3) DPCM on DC component

A technique called DPCM is instead applied on the DC value of each block. In practice it is possible to encode the DC component of a block as the difference compared to the value of the previous block because, generally, a statistical relationship in images among the DC components of adjacent blocks exists. This trick allows a further reduction of the space occupied by the data.

d4) Huffman compression

The classic variable code length coding is the last entropy coding applied to the data. The data is divided into 'words' (strings of bits); the statistical frequency of each word is analyzed and each word is recoded with a variable length code according to frequency of appearance. A short code for the words that appear frequently and progressively longer codes to less frequent ones. Altogether the number of bits needed to represent the data is reduced consistently.

Decompression

JPEG is a symmetrical code for its own nature and then the processing required for decompression is the exact inverse of that required for the compression.

The Huffman decompression is applied on compressed data; the resulting data are used for the reconstruction, block after block, of DC and AC components, then the coefficients are multiplied by an inverse quantization table. The resulting 8x8 block is subjected to an inverse DCT and at this point, depending on the settings of the file, the RGB image has already obtained, or you must do the YUV -> RGB conversion.

Conclusion

JPEG is a standard quite simple in its essence and yet the importance it has assumed in computing has been and is still really huge. Think of how many JPEG images are downloaded via the Internet every day, think of how many images are stored in digital cameras or digital video codec to be processed (many codec use the basics of JPEG compression to compress images sequences that form the video). All that would not have been possible without the commitment of the Joint Picture Expert Group. But as with many other technologies the JPEG is approaching to retirement age. Indeed, the new JPEG2000 standard promises better performance in compression, higher quality and more features in the security.

4.2 JPEG 2000

In the previous section we discussed one of the most popular and disseminated to the world compression formats, the JPEG. The mathematical principles which are the basis of the famous image compression format have been described.

JPEG is universally accepted as the standard 'de facto' in the field of lossy compression of images. Recall also that the basic mathematical tool in the JPEG (DCT) is used both in video compression of movies (MPEG, DiVX) and in audio compression (MP3).

However, in December 2000, the ISO / ITU has completed the standardization process and adoption of the new compression image algorithm, JPEG2000, which probably will become the successor to JPEG.

But how JPEG 2000 exactly works? What are the main differences from the older brother JPEG? Will JPEG 2000 definitively replace JPEG or will the glorious predecessor still have space?

How JPEG 2000 works

The differences compared to JPEG are numerous, indeed we can say that the two formats differ in almost all areas. The main evolution is certainly represented by the change of the basic mathematical tool at the base of the compression algorithm. While, as we explained previously, JPEG uses DCT (Discrete Cosine Transform) and operates on 8x8 blocks of pixels, JPEG 2000 uses the DWT (Discrete Wavelet Transform) and essentially operates on the entire image.

The word *wavelet* refers to the particular decomposition that is made of the image information. If the DCT decomposes the image into harmonic components (in practice is a frequency analysis) and block by block, the DWT breaks down the entire image into subbands a cascade. By adopting this approach and extending the analysis to the entire image, it eliminates the main drawback of JPEG: the excessive "tiling" that occurs with increasing the compression factor. This type of filtering (that will analyze in detail later) is obtained by the convolution of the image with the particular FIR filters, that remember 'small waves' for their structures.

A list of the main features provided by the JPEG 2000 specification is reported below:

- Support for different color spaces and modes (two-tone images, in grayscale, at 256 color, at millions of colors, in RGB standard, PhotoYCC, CIELAB, CMYK).
- Support for different compression schemes, adapted as needed.
- Open standard for subsequent implementations related to the emergence of new needs.
- Support for the inclusion of an unlimited amount of metadata in the space of the file header, used to provide private information or to interact with software applications (drive, for example, the browser to download appropriate plug-in from Internet).
- State of the art for the destructive and non-destructive images compression, with a saving of space at the same quality, compared to the JPEG standard, which starts from a +20-30%.
- Support for files larger than 64k x 64k pixels, or greater than 4 GB.
- Support for the data transmission in disturbed environments, for example through the mobile radio.
- Quality scalability (signal / noise ratio or SNR) and multi-resolution.
- Support for ROI (Region Of Interest) encoding, ie of areas considered most important and therefore saved with a greater resolution than that used for the rest of the image.
- Native support to the Watermarking or branding for the recovery of royalties.

Introduction to the Discrete Wavelet Transform

JPEG2000 uses the DWT (wavelet) transform in the main compression stage. The Wavelet transform decomposes gradually the original image in sub-band representations at multi-resolution. To understand how it works first consider the one-dimensional case.

A certain X signal is first split into two parts by passing it through a high pass filter (A_H) and a low pass filter (A_L). The two signals thus obtained are decimated by 2 (ie a sample is taken every 2) giving rise to two subbands X_H and X_L . If initially we had an X signal of N samples, now we have two signals X_H and X_L each consisting of $N/2$ samples. Generally they are the first order coefficients of the wavelet transform; X_H is also called detail..

If the filters used respect certain characteristics, the process is reversible and it is possible to extract the original signal from the two components interpolating the coefficients and by passing them through appropriate filters. The low-frequency sub-band that typically contains the most important part of the signal, can be further decomposed a cascade as shown in the figure:

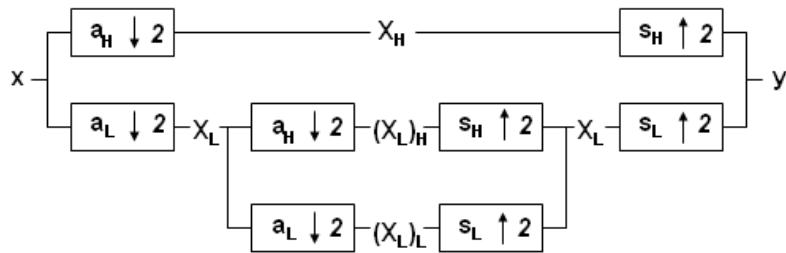


Figure 4.3 Decomposition of low-frequency sub-band.

In the two-dimensional case of an image, the signal (the image) is decomposed into sub-bands through three filters: a horizontal high-pass filter, a vertical high-pass filter and a diagonal high-pass. The resulting images are decimated both horizontally and vertically to form three groups of detail coefficients (HL, HH, LH). The original image is also scaled of a 2 factor on X and Y and forms the LL sub-band in low frequency. This particular sub-band is then and repeatedly broken down at successive levels of decomposition (see image).

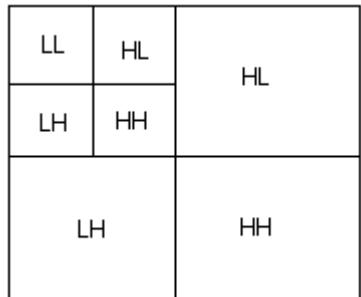


Figure 4.4 JPEG 2000 decomposition levels.

This type of representation with the use of DWT has many advantages that will see later.

JPEG 2000 in detail

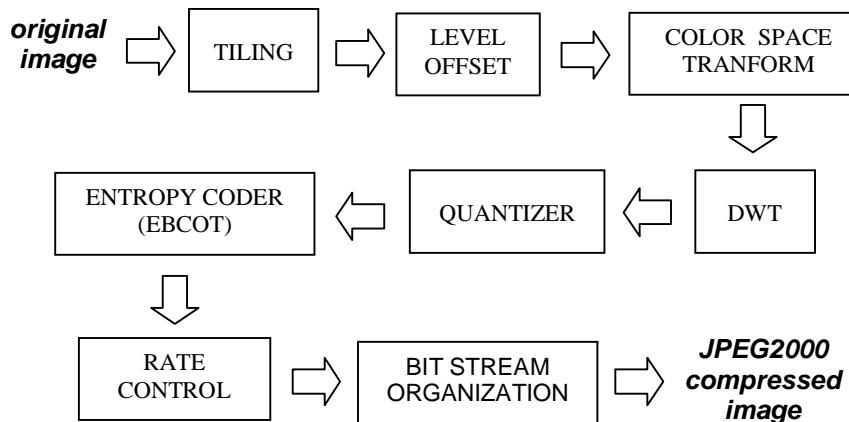


Figure 4.5 JPEG 2000 encoder architecture.

The JPEG2000 format expects several image processing steps; below we will explain the fundamental algorithms without going too deeply into mathematical issues and the numerous exceptions provided by the algorithm.

a) Pre-processing

The original image is first divided into non-overlapping tiles of equal size. Each tile is separately compressed and the compression parameters can be different from tile to tile.

The unsigned values that describe the pixels are properly shifted by a constant factor in order to be centred in 0. In addition, each tile is separated into color channels. Two basic color coding were defined in JPEG 2000, one reversible for lossless compression and the other one irreversible for lossy type and perfectly analogous to the conversion from RGB to YCbCr used in JPEG.

It is also possible to compress images in GrayScale and at levels.

b) DWT

The DWT operates over the whole image and causes a subsequent breakdown in sub-bands, each which treatable by importance taking in human vision. In the following images we can see the various levels of decomposition.



Figure 4.6 First DWT decomposition level.

At the first level, the original image was split into 4 sub-images (each with a quarter resolution of the original) that represent the details in high frequency and the content in low frequency.

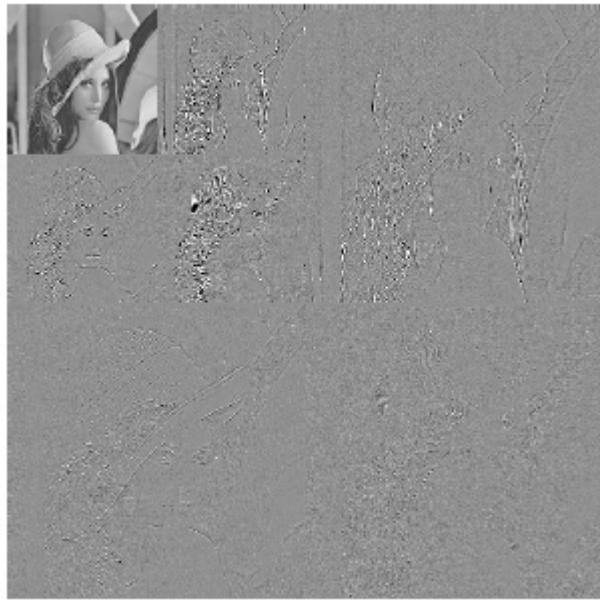


Figure 4.7 Second DWT decomposition level.

The second level of decomposition operates on low frequency content generated by the first level. Other detail subbands and a new low-frequency subband are generated.

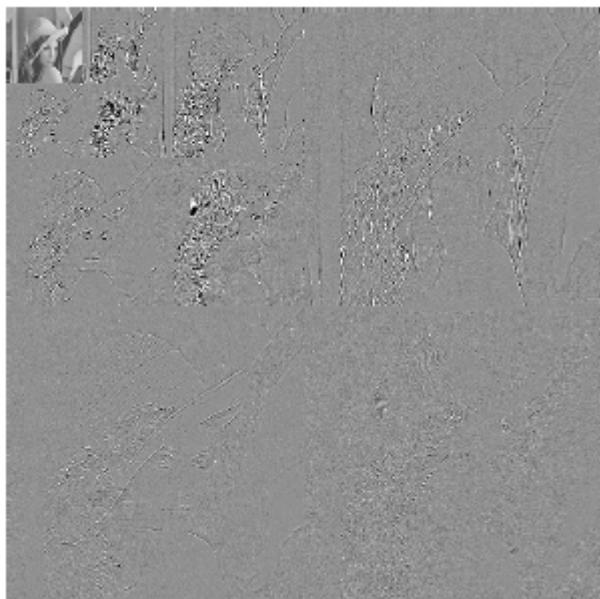


Figure 4.8 Third DWT decomposition level.

The process proceeds to successive levels by generating more and more detail subbands. The details generated at successive levels of decomposition are more important for the perceptual purposes of those generated in the early stages.

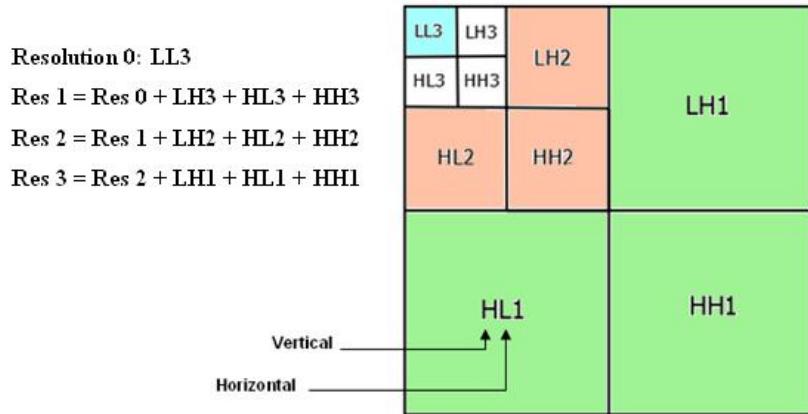


Figure 4.9 Resolution levels.

The filters used for the decomposition into sub-bands are called bi-orthogonal filters for their mathematical characteristics. Here are some examples of finite impulse response (FIR) filters used for the extraction of sub-bands; their characteristic gives the name to the wavelet encoding.

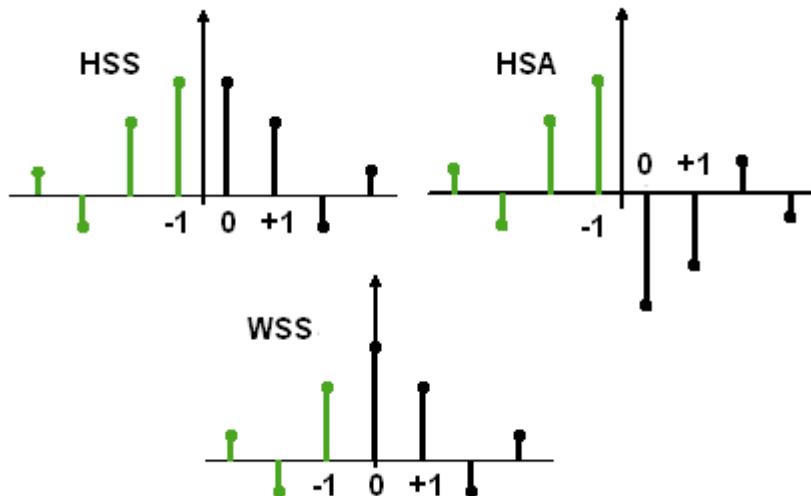


Figure 4.10 FIR filters.

c) Quantization

The behaviour of the human eye compared to the spatial frequency of the images may be determined by the response to harmonic images. It is possible to draw a function of the sensitivity of the human eye to the contrast changes. This sensitivity is maximal for images that are 5 harmonious cycles for viewing corner, while vanishes almost completely for images with 50 cycles for corner.

This feature allows us to define the appropriate quantization values for the relevant sub-bands. It is clear that the first sub-bands of detail can be quantized with accuracy much smaller than in later ones since the human eye can not perceive low entity changes at high spatial frequencies. The exponential decreasing shape of the curve also explains the reason of the decimation which is performed at each iteration.

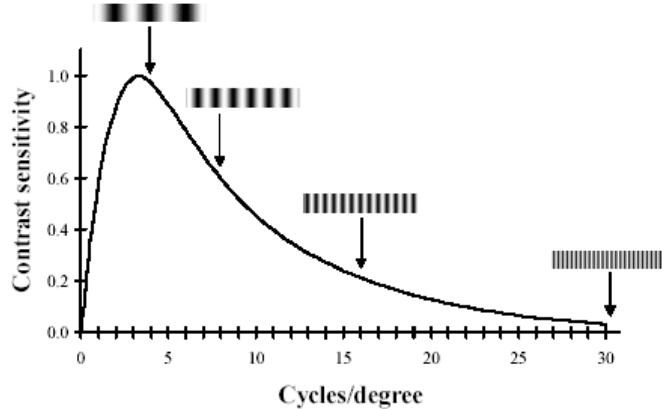
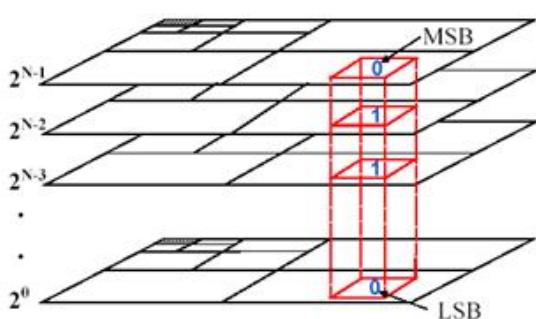


Figure 4.11 Contrast sensitivity curve.

d) Entropy coding

The bits uniformly quantized in the quantization stage are handled in bit-plane. This aggregation of data results in a substantial inefficiency of the Huffman entropy coding (referred to JPEG case). To ensure efficient entropy coding is necessary to merge pattern of bits in configurations at high probability.

The "arithmetic coding" provide excellent performance in these cases and allow to deviate from the theoretical peak performance of just 5-10%. The arithmetic codes used in the JPEG2000 are also adaptive and estimate the symbols probability based on adjacent pixels values adapting to the image content.



The entropy coding takes place for the code-block (eg 8x8 pixels). The bit planes of each block are examined in succession from MSB (Most Significant Bit) to LSB (Least Significant Bit).

Figure 4.12 Embedded Quantization by Bit-Plane Coding.

This approach allows to decode the image, bit plane by bit plane, in order to give the decoding scalable in terms of signal to noise ratio (SNR).

e) Generation of the final bit-stream

In the previous step a bit-stream for each code-block was generated; in this phase the bit-stream is organized to be able to easily handle the scalability in quality (SNR) and resolution. This aim is achieved by packing the data for bitplane and then for decomposition levels. In the decompression phase, it is possible to use only a certain number of bitplanes (from the MSB onward) depending of the desired quality. If a certain resolution is required, it is only necessary the treatment of subbands of higher order.

The two approaches can be combined. This feature of scalability and flexibility of the JPEG 2000 algorithm is one of the most interesting. The application servers will be able to provide services based on JPEG2000 managing the resolution and quality according to the application characteristics and the available bandwidth. JPEG 2000 will be the first step toward new video compression techniques in which we will have a single file to satisfy the most stringent requirements in terms of resolution, quality and, ultimately, the request band.

The final structure is not simple at all and then you just know that it is designed not only for scalability but also to satisfy two other interesting requirements: the management of ROI (Region Of Interest) and error robustness.

The ROI coding is the ability to encode different parts of the image with different parameters and different qualities. The choices can be made both in the compression and in content delivery phase and in this case managed from a server according to the user needs.

JPEG 2000 also provides strategies for using the format in disturbed channels with minimal loss of functionality. Unlike JPEG in which the loss of data determines the irremediable corruption of video blocks, the robust JPEG2000 encoding allows to ensure high standards of quality even on noisy channels. It will be particularly appreciated on cellular systems as well as for streaming systems in broadcast where the loss of packets is, in this way, tolerable.

The color transform

The JPEG 2000 supports images in multiple components, ie images made up from different color channels. Each component is constituted by a matrix of values which, in conjunction with other color components, forms the final image. The different components can have different bit depth, as well as it is possible that some components are represented in sign while other not.

Some transformation on the image components are carried out before the real coding. The possible transformations are two: one irreversible (ICT) and the other one reversible (RCT).

Generally, an image is represented in terms of its chromatic components. red, green and blue (RGB). The coefficients of each component are shifted, before the transformation process. The figure shows a diagram in which the entire sequence of mentioned operations is representing.

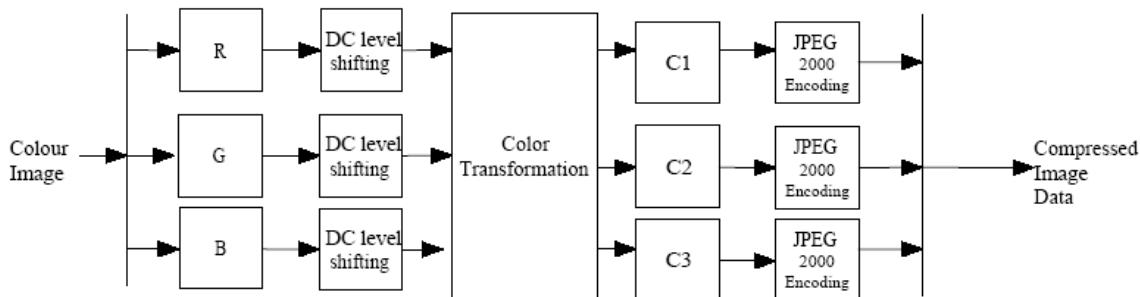


Figure 4.13 Color transformation.

The ICT transformation can be carried out only in the case of lossy compression. ICT transformation and its inverse are made by the (1) and (2) equations.

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.5 \\ 0.5 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.0 & 0.0 & 1.402 \\ 1.0 & -0.344 & -0.714 \\ 1.0 & 1.772 & 0.0 \end{bmatrix} \left(\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} - \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} \right) \quad (2)$$

The RCT transformation can be used both for lossy and lossless encoding.

This type of transformation allows to achieve three results:

- color de-correlation for a better compression,
- the use of a color space closer to the human visual system,
- the possibility to get a lossless compression.

The RCT and its inverse transform is made through the (3) and (4) equations.

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} \left\lfloor \frac{R + 2G + B}{4} \right\rfloor \\ R - G \\ B - G \end{pmatrix} \quad (3)$$

$$\begin{pmatrix} G \\ R \\ B \end{pmatrix} = \begin{pmatrix} Y - \left\lfloor \frac{U + V}{4} \right\rfloor \\ U + G \\ V + G \end{pmatrix} \quad (4)$$

Strengths of the new algorithm

In summary, we list the main strengths of the new algorithm format:

- Increase of encoding efficiency of at least 30%, proportionally to the increase of the compression factor.
- Representation in multi-resolution.
- Scalable Quality (SNR), from the loss-less to very low bitrates.
- Presettable bit-rate for application at fixed bit-rate
- Error transmission robustness
- Native support to Watermarking.

Comparison with JPEG

Finally we compare the new algorithm with the more traditional JPEG. There is first to say that the JP2 (this is the extension of JPEG 2000) does not quickly take over its predecessor, for several reasons. For low compression ratios, JPEG retains a certain efficiency and the new features of the JPEG2000 probably will not justify the transition.

It should be noted that the new format is much heavier to handle if compared to the JPEG and this may be a critical parameter for systems such as digital cameras, hardware for special applications and so on.

If the compression ratio rises, the previous consideration changes radically. If JPEG 2000 is more efficient of only 20-30% for compression factors in the range from 1:10 to 1:30, in the higher range the efficiency increases exponentially. At 1:100 or higher factors, an image compressed with JPEG 2000 is still usable in contrast to a completely unrecognizable JPEG image.

The image below is a demonstration of what has been said. An original quality image has been compressed with a 105:1 compression factor. The upper image is a JPEG and the bottom one is a JPEG 2000.

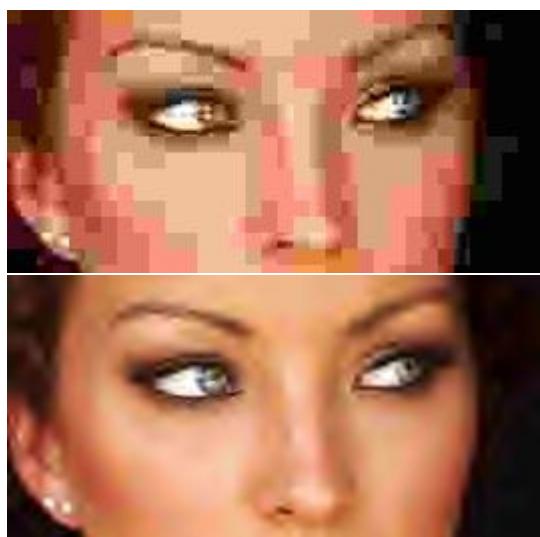


Figure 4.14 Comparison between JPEG and JPEG 2000 images.

This high capacity to maintain image intelligibility at high factors can make the JPEG 2000 to be extremely useful on very narrow channels as the wireless ones, or for multimedia plug-in such as the FlashPlayer, or still for the management of movies in narrow-band. The main problem remains the superior computing power and the amount of memory needed for compression and decompression.

4.3 JPEG XR

The JPEG XR is a novel still-image compression algorithm recently standardized by ITU-T and ISO/IEC standard and is based on the technology originally developed by Microsoft under the name HD Photo. It addresses the needs of a broad range of consumer electronics applications, particularly including digital photography.

The standard JPEG XR is defined in a document called "Information Technology - JPEG XR Image Coding System" (ISO / IEC 29199) consisting of five parts:

- **Part 1:** System Architecture (ISO / IEC 29199-1): This part outlines the specifications and the encoder and decoder design.
- **Part 2:** Image coding specification (ISO / IEC 29199-2): This part describes in detail the encoding format of this standard.
- **Part 3:** Motion JPEG XR (ISO / IEC 29199-3): This section describes the use of the JPEG XR encoding for storing moving images sequences. This format is based on the ISO Base Media File Format.
- **Part 4:** Conformance testing (ISO / IEC 29199-4): This section shows a series of tests to be used to verify if the different encoders, decoders, files, code streams are in conformity to the document specifications [Part 2]
- **Part 5:** Reference Software (ISO / IEC 29199-5): This part provides the reference software for Part 2, also used for the construction of new encoders and decoders and to perform conformance testing and interoperability. [5] [6] [7] [8] [9] [10].

JPEG XR algorithm

The structure of JPEG XR image compression algorithm is based on a block transform which, as shown in Figure 4.15, follows a classical organization scheme: color conversion, transform coding, quantization, and entropy coding.

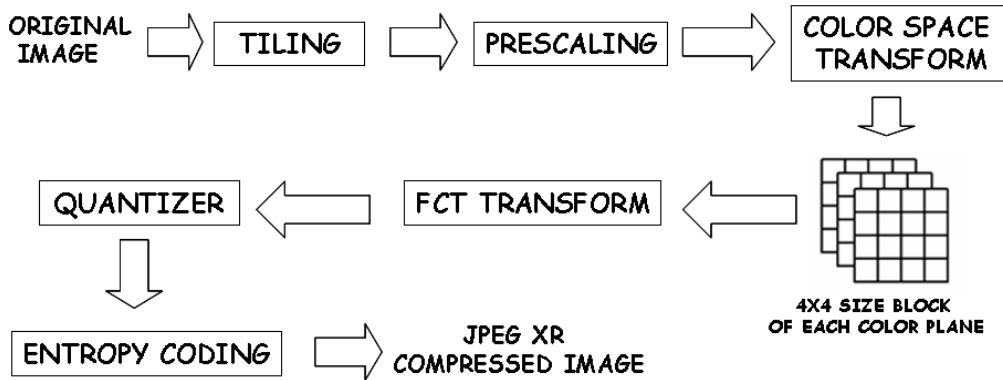


Figure 4.15 JPEG XR encoder architecture.

First, the source image is partitioned into rectangular non-overlapping blocks and then it is converted from RGB to YIQ (or YUV) color space. Although this separation is not strictly necessary, allows a better image compression.

In order to be processed, the image is divided in color plane, one for each color channel, and within each plane it is again divided into 4x4 blocks.

The processing key is undoubtedly the FCT transform in its two-dimensional version, which will be explained below.

After the FCT transform, the quantization process begins. Quantization allows us to eliminate the less important information. Now it is necessary to adopt a set of empirical techniques to reduce the amount of image needed to transmit the remaining significant information.

4.3.1.1. Pre-scaling

The pre-scaling steps are usually used for input data range greater than 27/24 bits, while they are optional for 16-bit unsigned and signed integer and 32-bit signed integer.

On the encoder side, the input data is right-shifted by some m bits to reduce the data range to 27/24 bits or below. When data is scaled 27 bit limit is used, while data is unscaled 24 bit limit is applied.

4.3.1.2. Color conversion

The encoder can perform color conversion from external RGB formats to internal YUV formats; a reverse process is applied at the decoder side.

The direct conversion from RGB to YUV444 is:

$$V = B - R$$

$$U = R - G + \left\lfloor \frac{V}{2} \right\rfloor$$

$$Y = G + \left\lfloor \frac{U}{2} \right\rfloor$$

The reverse conversion from external cmyk to internal YUVK is:

$$V = c - y$$

$$U = c - m - \left\lfloor \frac{V}{2} \right\rfloor$$

$$Y = k - m - \left\lfloor \frac{U}{2} \right\rfloor$$

$$K = k - \left\lfloor \frac{Y}{2} \right\rfloor$$

If the color format is YUV422 or YUV420, before performing the downsampling, the encoder first performs a conversion to an intermediate YUV444. The corresponding over-sampling is carried out through the decoder.

If any sub-sampling or over-sampling is included, color conversion is perfectly reversible and no information is lost in the conversion of color.

4.3.1.3. Hierarchical image partitioning

The spatial hierarchy of a JPEG XR image, as shown in the figure below, consists of 5 layers:

1. Pixel, that is an integer corresponding to a color channel in a specific location
2. Block, that is a 4x4 matrix of adjacent pixels related to the same color channel
3. Macroblock, that is a 4x4 matrix of adjacent blocks which includes both luma and chroma components.
4. Tiles, set of macroblocks corresponding to particular regions of image plane
5. Image

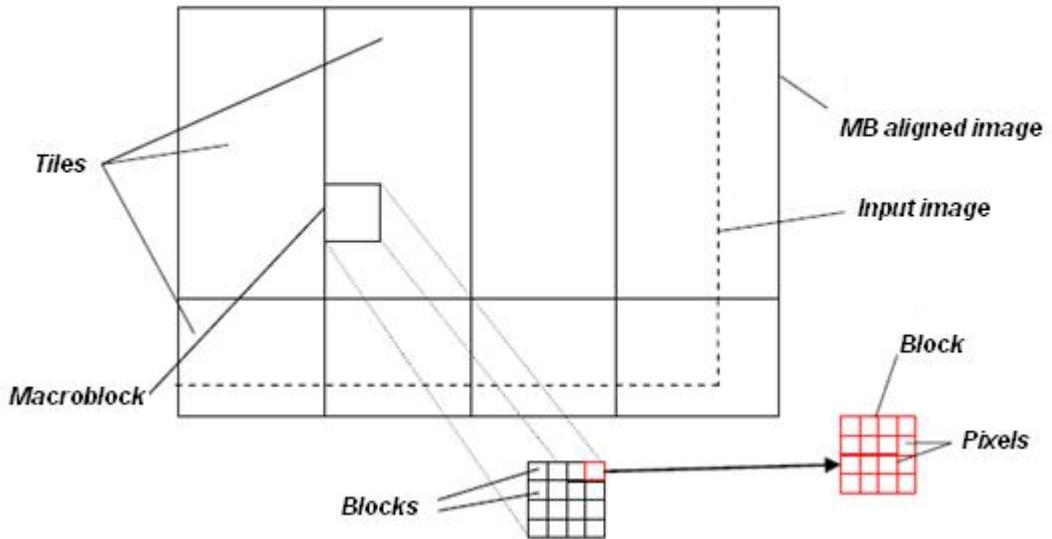


Figure 4.16 Hierarchical JPEG XR image partitioning [6].

4.3.1.4. Structure of the Bit-Stream

The JPEG XR defines two approaches for access to the bit-stream: spatial and frequency. In both case, the bit-stream is composed by an image header followed by progressive tiles. In the first case the bit-stream of each tile is arranged in a macroblock order; in the second case the bit-stream of each tile is transmitted in multiple tile packets. In the frequency mode layout the bit-stream of each tile is set up as a hierarchy of bands, as shown in Fig.4.17.

The tile coefficients are positioned in the following order: DC, LP, HP and FLEX band. FLEX provides additional information to HP band. The FLEX band may be not present.

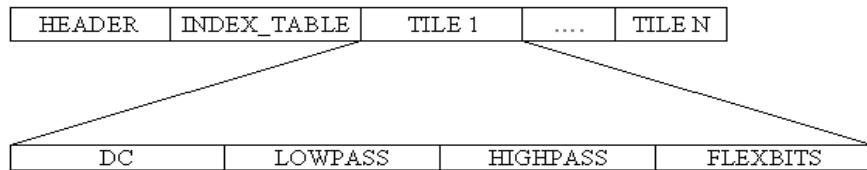


Figure 4.17 Layout of JPEG XR bit-stream in frequency mode [4].

The group formed by DC, LP and HP sub-bands, produces different resolutions of the image information; while FLEX, if exists, can be used for progressive decoding.

The first scale of decoding can be obtained decoding the part of bit-stream that corresponds to DC, LP and HP coefficients.

It is then possible decode FLEX to produce the complete image.

A resolution equal to 1:16 is obtained decoding only the DC band. To obtain a 1:4 resolution of the image, only DC and LP bands need to be decoded.

4.3.1.5. JPEG XR transform

JPEG XR coding algorithm [4] is based on a two-level hierarchical lapped transform [6], that is a concatenation of two transform operator: the FCT (Forward Core Transform) and the OT (Overlap Transform).

In order to execute the transformation, an image JPEGXR is divided into 4x4 macroblocks, for each color plane. Every of them consist of 16 4x4 non-overlapping blocks, on each of which is applied the FCT transform, that produces 1 DC and 15AC coefficients of first stage for any block.

The DC coefficients of all blocks are grouped together in a 4x4 block and then a FCT transform is applied again.

So, this produces other 16 coefficients: 1 DC and 15 AC coefficients of second stage. Finally, this coefficients are mapped in the first pixel of each block that is a part of the macroblock. Fig. 4.18 shows the coefficients mapping of the two stages of FCT, that produce 240HP coefficients of first stage and 15LP and 1 DC coefficients of second stage.

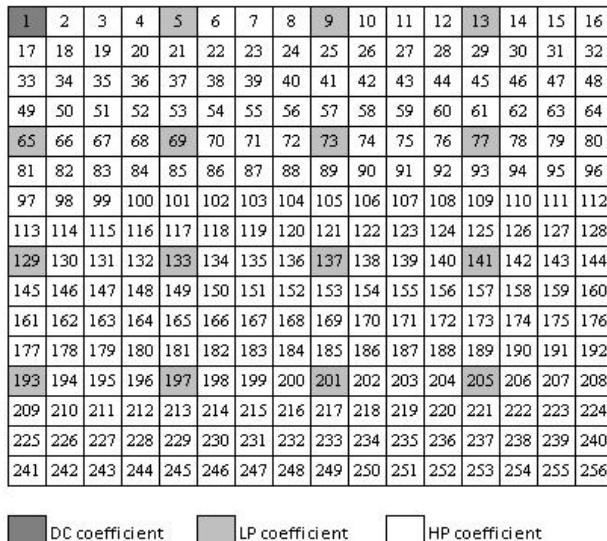


Figure 4.18 JPEG XR macroblock coefficient mapping.

The FCT Transform, the core of the transform, consists of three elementary 2x2 filter operations:

- 2x2 Hadamard Transform: T2x2h;
- 1D rotate: TOdd;
- 2D rotate: TOddOdd.

FCT is applied to each 4x4 block in two different stage, as shown in Figure 4.19. Each stage provides four 2x2 transform which may be done simultaneously or in a random sequence inside the stage. Anyhow, the second stage transform can initiate only if the first stage transform is completed.

The Hadamard matrix [5] (with elements values either +1 or -1) of second order is given by:

$$H_{2,2} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

while a Hadamard matrix of order 2^j can be written as

$$H_{2^j, 2^j} = \begin{bmatrix} H_{j,j} & H_{j,j} \\ H_{j,j} & -H_{j,j} \end{bmatrix}$$

We can find Hadamard matrices of order other than powers of 2 , but they are not common in image processing. [6]

Inverse Hadamard matrix are easily computed as

$$H^{-1}_{j,j} = \frac{1}{j} H_{j,j}$$

The Hadamard Transform and its inverse are given by:

$$F = H_{M,M} f H_{N,N}, \quad f = \frac{1}{MN} H_{M,M} F H_{N,N}$$

where f represents the original image and F the transformed image.

The benefits of Hadamard transform are that the elements of its matrix are binary and real numbers, and the rows or columns are orthogonal.

The 2x2 Hadamard Transform is given by this simple steps:

```

e = a + d
f = b - c
T1 = (e - f +
ValRound)/2
T2 = c
C = T2 - d
D = T1 - T2
A = e - D
B = f + C

```

where ValRound is a factor that can assume only the value 1 or 0; a,b,c,d are the original values and A,B,C,D are the values resulting from the transform process.

The 2-point rotation operator, necessary for the TOdd and TOddOdd operations, is given by:

$$T_R = \begin{bmatrix} -\cos\frac{\pi}{8} & \sin\frac{\pi}{8} \\ \sin\frac{\pi}{8} & \cos\frac{\pi}{8} \end{bmatrix}$$

The product of Kronecker [7] is an operation between two matrix of arbitrary size resulting in a block matrix ; i.e if A is a matrix $(m \times n)$ and B a matrix $(p \times q)$, their product of Kronecker

$$A \otimes B = \begin{bmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \dots & a_{mm}B \end{bmatrix}, \text{ returns a matrix } (m \times p)(n \times q).$$

Now, we can define the operator TOdd as a product of Kronecker between T_H and T_R , where $T_H = H \otimes H$ (H is a Hadamard matrix); while the operator TOddOdd is the product of Kronecker between T_R and itself.

$$T_{Odd} = T_H \otimes T_R$$

$$T_{OddOdd} = T_R \otimes T_R$$

The Todd operator is defined by this code:

```
f = b - c
e = a + d
g = c + ((f+1)/2)
h = ((e+1)/2) - d
T1 = f - ((3*e + 4)/8)
T2 = e + ((3*f + 4)/8)
T3 = h - ((3*g + 4)/8)
T4 = g + ((3*h + 4)/8)
D = T3 + (T1/2)
C = T4 - ((T2+1)/2)
B = T1 - T3
A = T2 + T4
```

Finally the TOddOdd operator is the result of:

```
f = - b
g = - c
h = d + a
T1 = h/2
T2 = g/2
V1 = g - f
e = a - T1
V2 = f + T2
V3 = e + ((V2*3 + 4)/8)
V4 = V2 - ((V3*3 + 3)/4)
V5 = V3 + ((V4*3 + 3)/8)
D = V4 - T1
C = V5 + T2
B = V1 + V4
A = h - V5
```

The first stage of FCT includes four 2x2 Hadamard Transform (T2x2h): earlier is applied to corners (a, b, c, d) and then to the centers coefficients (e, f, g, h) of a 4x4 block; afterwards the T2x2h is applied to upper and lower edges (i, l, m, n) and finally to the right and left edges (o, p, q, r).

The second stage continues with a T2x2h for even-even basis (A, B, C, D), with a 1D rotation for even-odd basis (E, F, G, H) and odd-even basis (I, L, M, N) respectively and lastly with a 2D rotation for odd-odd basis (O, P, Q, R).

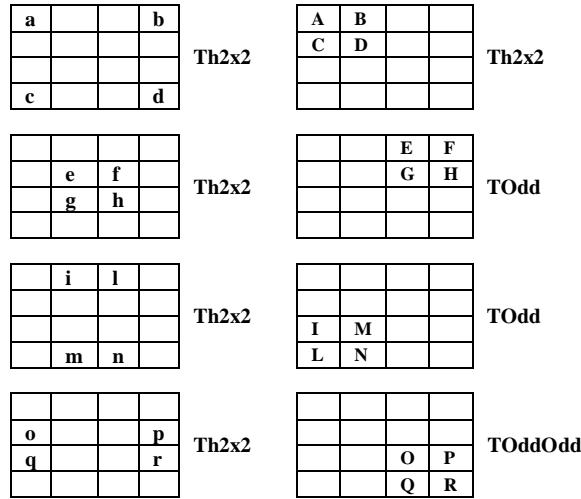


Figure 4.19. Forward Core Transform steps.

After the two stages, coefficients are re-ordered as shown in Fig.4.20.

i	Array[i]	i	Array[i]
0	0	8	1
1	8	9	11
2	4	10	15
3	6	11	13
4	2	12	9
5	10	13	3
6	14	14	7
7	12	15	5

Figure 4.20. Forward Permutation.

Every FCT operation is preceded by an optional filter OT (Fig.4), which is applied to a 4x4 areas, between the stacked two-dimensional blocks. (If OT_mode=0, no overlap operator is applied; if OT_mode=1 only the first level overlap is applied, otherwise if OT_mode=2 both level overlaps are performed). The overlap filter is designed to limit the blocking artifacts.

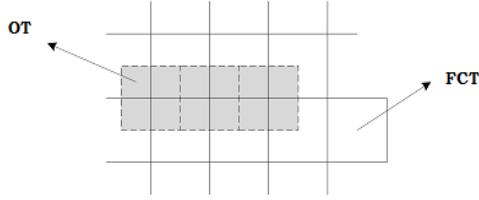


Figure 4.21 Region of support for 4x4 FCT and OT operators [4].

The JPEG-XR decoder uses a block transform, called ICT, in which the stages are inverted. Every filter operations within the stages use its own inverse transform, and it is preceded by the inverse permutation function. The decoding process is summarized in Fig. 5.

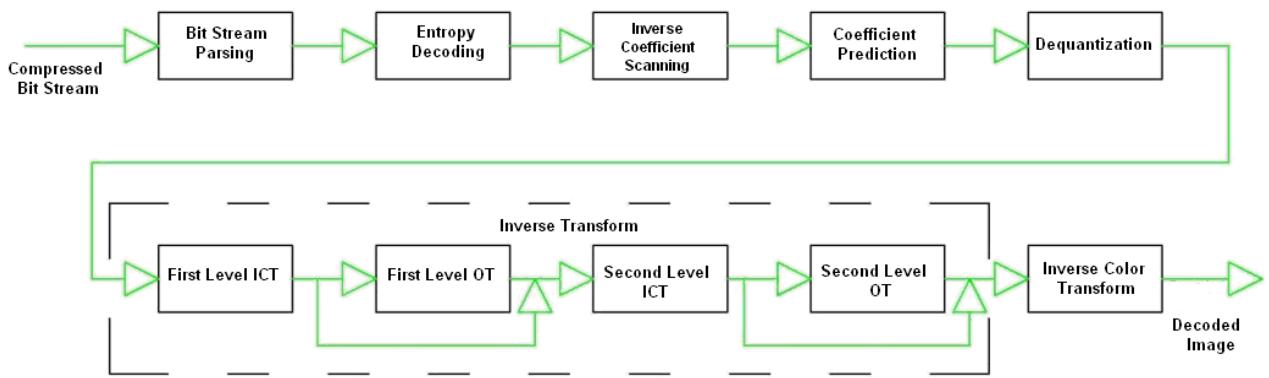


Figure 4.22. Decoding process diagram [5].

4.3.1.6. Quantization

The process of quantization allows to reduce the entropy of each coefficient obtained from the FCT transform, by dividing them for an appropriate value, called the *quantization value* (QP).

The choice of this parameter is fundamental to obtain a good compromise between the compressed image quality and the number of bits necessary to encode the image.

For lossless encoding, the quantization value must be equal to 1; for lossy compression it is possible to choose a quantization value greater than 1.

JPEG XR standard offers three quantization parameters (QP) types: color plane, frequency and spatial region.

4.3.1.6.1. *Quantization in the spatial dimension*

JPEG XR enables the following flexibility types:

- The same QP value can be used to encode the entire frame,
- The same QP value can be used to encode the entire tile, while different tiles within an image can use different QP
- Different values of QP can be used from different macroblocks within a tile

4.3.1.6.2. *Quantization across frequency bands*

JPEG XR also allows flexibility across the frequency bands, by varying the quantization rules.

- The same QP value is used from all frequency bands
- DC and HP bands use the same QP value, while LP uses a different value
- LP and HP bands use the same QP value, while DC uses another value
- Each band has a corresponding QP value

4.3.1.6.3. *Quantization cross color planes*

The connection between the QP value of various planes can be defined as follows:

- QP is identical for all color planes (uniform mode)
- QP value for the luma color planes is different to this for all other color planes (mixed mode)
- Each color planes can have a different QP value (independent mode)

4.3.1.7. Prediction

The JPEG XR standard uses the adaptive prediction of the transform coefficients to improve coding efficiency. In fact, it exploits the similarity between two close macroblocks to determine the value of their respective coefficients. It is called adaptive because the prediction changes depending on the similarity of nearby macroblocks to the reference one.

There are several predictive algorithm which work for example on the macroblocks that are located above, below, left or right to the reference one and each time make reference to the most similar.

JPEG XR provides three types of prediction such as:

- DC prediction (prediction of DC values of first level transform)
- LP prediction (prediction of DCAC values of second level transform)
- HP prediction (prediction of DCAC values of first level transform)

Prediction is used only if there is a strong and dominant orientation in inter-block correlation. If tiling is used, each tile is considered as a separate image for ensure an independent decoding of tiles.

4.3.1.7.1. DC prediction

The DC coefficient of a macroblock can be predicted in one of the following directions: TOP predictor = DC [L] , LEFT (predictor = DC [T]), both together (predictor = DC [L] + DC [T] $\gg 1$) and no prediction (predictor = 0).

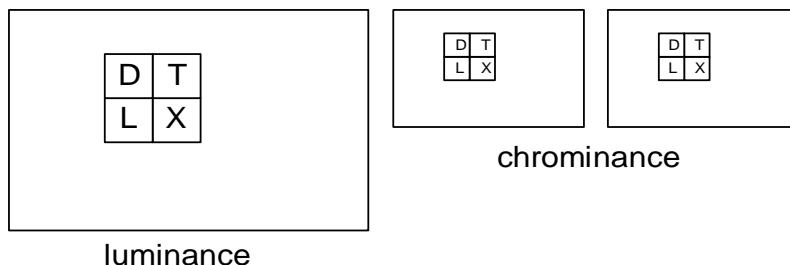


Figure 4.23 DC prediction [12].

In this prediction mode a comparison between the DC values of the top and top-left boundaries and those of the left and top-left neighbours. If the difference in that direction is much smaller, it is possible to use the DC prediction from LEFT or from the TOP.

4.3.1.7.2. LP prediction

The three modes allowed in LP prediction are:

Prediction of the first column from left (predictor = LP [L])

Prediction of the first row from top (predictor = LP [T])

No prediction (predictor = 0)

LP prediction depends on DC prediction mode, the quantization value and its predictor. The LP coefficients are predicted from left if the DC is predicted from the left macroblock and its QP is equal to the QP of the current macroblock. The LP coefficients are predicted from top if the DC is predicted from the top macroblock and its QP is equal to the QP of the current macroblock. Otherwise there is no prediction for the LP coefficients.

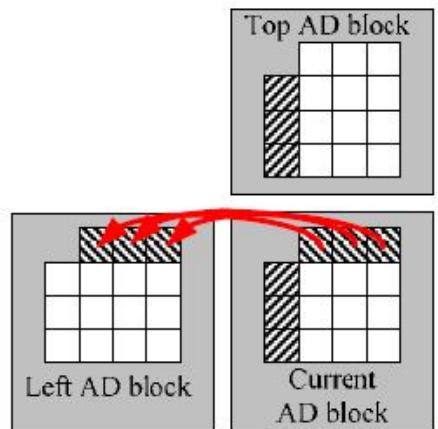


Figure 4.24 LP prediction [11].

4.3.1.7.3. HP prediction

The modes allowed for HP prediction are equal to those of LP prediction. HP prediction of a block is executed only from the blocks within the same macroblock. Otherwise no prediction is performed for that block.

HP prediction depends on LP coefficients values previously predicted. HP prediction from top is chosen if the energy of the first column of LP coefficients is much smaller than the energy of first row of LP coefficients. Vice versa the HP prediction from left is performed. Otherwise no prediction is done.

HP prediction from left is shown in Figure 4.25:

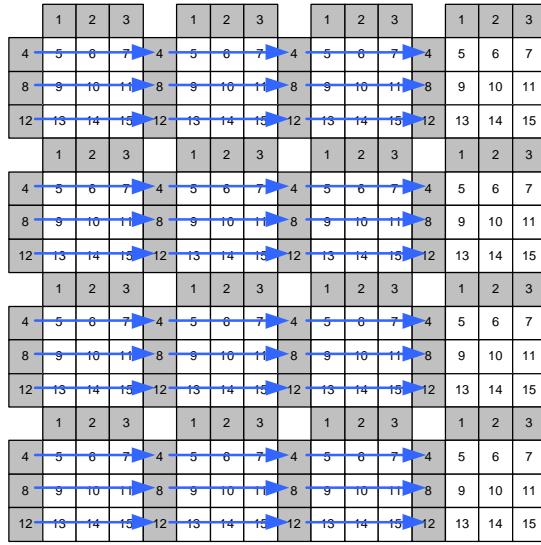


Figure 4.25 Highpass prediction from left [12].

4.3.1.8. Adaptive orders for coefficient scanning

The objective of the coefficient scanning is to convert the various matrices of the coefficients obtained from the transformed into one-dimensional arrays on which to apply the entropy coding.

A difference of the JPEG standard classic that uses a zig-zag scanning, JPEG XR uses an adaptive scanning based on a local statistic of the previous coefficients. The adaptation process arranges the scan pattern in order to earlier scan the coefficients with higher non-zero values probability.

JPEG XR uses three scan patterns: *lowpass*, *highpass horizontal* and *highpass vertical*. The first is used for the lowpass transform coefficients in a macroblock, while the other two are used for the highpass transform coefficients in a macroblock. The three scan pattern are initialized to a specific ordering at the start of each tile and are dynamically adapted.

This algorithm provides the use of two arrays:

- *Order [i]*, which contains the order of the current scan. For example, order [3] = 5 means that the coefficient number 5 has to be scanned for third.
- *Counter [i]*, which contains the number of nonzero coefficients acquired before the current block. For example Counter [3] = 24 means that prior to scanning of this block 24 coefficients different from zero in the same position of the coefficient 5 were found.

If the *order [i]*-th coefficient has a non-zero value the associated *counter [i]* element is incremented by 1. *Counter* checks if the current coefficient occurs more frequently than the

previous in the scan and if this condition is satisfied *Order* array is adjusted by swapping the relative positions of the above coefficients.

Scan Order	4	1	5	8	2	9	6	12	3	10	13	7	14	11	15
Scan Counter	32	30	28	26	24	22	20	18	16	14	12	10	8	6	4

Figure 4.26. Example of the initial array for the LP coefficients [13].

Scan Order	1	4	5	8	2	9	12	6	10	3	7	13	14	15	11
Scan Counter	33	32	31	28	24	22	21	20	17	16	13	12	11	7	6

Figure 4.27. Array of LP coefficients after three scans [13].

The arrays are initialized at the beginning of the image (top left).

The *counter* arrays, both for LP and HP coefficients, are always initialized with a constant array of descending values. The *counter* arrays are reset at every eight macroblocks and at the start of every tile, while the *order* arrays are reset only at the start of tiles

4.3.1.9. Entropy coding

To reach a greater efficiency the array obtained from quantization process is subjected to the entropy coding procedure, which allows compression by using some special mathematical algorithm. These algorithms are essentially based on the possibility to make more compression in areas that have more occurrences of the same symbols in the near position. One of most widely used is the RLE (Run-Level Encoding), in which sequences of consecutive identical symbols are combined with each other and represented by a single symbol or a code. The symbol represents the runs of zeros between nonzero transform coefficients and the values of nonzero coefficients.

Coefficient coding in JPEG XR is quite different. First, the coefficients are normalized and the bits resulting from the normalization are coded as fixed length FLEXBITS. The joint symbols indicates the value of nonzero coefficients with the run of zeros after that coefficient. The size of VLC tables is reduced in order to minimize the memory usage. VLC table are adapted according to the local statistic of previously coded symbols.

4.3.1.9.1. Adaptive VLC table

JPEG XR uses adapted VLC (Variable Length Coding) tables, which assigns shorter code words to more frequent symbols in order to approximate the source entropy.

VLC tables adaption may be made in one of two ways:

- **Forward adaption**, in which the VLC table is created from the predefined set and then explicitly signalled in the bitstream. In fact, it is based on a first-pass through the data or using heuristics. This method, because of the significant signalling overhead, can be used only rarely.
- **Backward adaption**, in which the code table is adjusted “on the fly” using statistics of previously coded symbols. There is no signalling overhead to specify the VLC table, but this arrangement provides high computational complexity.

JPEG XR solves this problem using a new VLC adaptation procedure, in which only a small number of representative VLC tables is predefined and designed for a wide range of statistics. During the entropy process, the most appropriate code table is selected based on the history of recently decode symbols.

If VLC table T_i is currently used for entropy coding, the transition choices are limited to the following options: T_{i-1} table or T_{i+1} table.

Two D_1 and D_2 “discriminants”, initialized to zero, estimates the advantage of transitioning to the T_{i-1} or T_{i+1} table. For example, D_1 is incremented in relation to the efficiency gain if a symbol s is more efficiently encoded in T_{i+1} than T_i . Otherwise it gets decremented.

VLC tables used in JPEG XR for entropy coding are shown in Figure 17.

A relative small number of tables is sufficient to provide adaptivity to a wide range of image statistics and compression ratios. Sometimes only two code table are present and so there is only one possible transition. If only one discriminant is required the adaptation complexity is further reduced. It can be concluded that the complexity of the adaptive VLC table algorithm is smaller than the complexity of the actual VLC encoding/decoding algorithm.

SYMBOL	Code 0	Code 1	Code 2	Code 3	Code 4
0	0000 1	0010	11	001	010
1	00 0001	00010	001	11	1
2	000 0000	00 0000	000 0000	000 0000	000 0001
3	000 0001	00 0001	000 0001	0 0001	0001
4	0 0100	0011	0 0001	0 0010	000 0010
5	010	010	010	010	011
6	0 0101	0 0011	000 0010	000 0001	0000 0000
7	1	11	011	011	0010
8	0 0110	011	100	0 0011	000 0011
9	0001	100	101	100	0011
10	0 0111	0 0001	000 0011	00 0001	0000 0001
11	011	101	0001	101	0 0001

SYMBOL	Code 0	Code 1
0	010	1
1	0000	001
2	0010	010
3	00001	0001
4	00010	00001
5	1	011
6	011	00001
7	0 0011	000 0000
8	0011	000 0001

SYMBOL	Code 0	Code 1
0	01	1
1	10	01
2	11	001
3	001	0001
4	0001	00001
5	00000	00000
6	0 0001	00000

SYMBOL	Code 0	Code 1
0	1	01
1	0000	0000
2	001	10
3	0 0001	0001
4	01	11
5	0001	001

SYMBOL	Code 0
0	10
1	001
2	0 0001
3	0001
4	11
5	010
6	0 0000
7	011

All code tables used in HD Photo for coefficient and coded block pattern coding, enumerated in binary

Figure 4.28 VLC tables used by JPEG XR [14].

4.3.1.10. Decoder features

JPEG XR enables many useful decoder features thanks to its structured data layout and bitstream format, frequency hierarchy and tiles. Some of these properties are described below.

4.3.1.10.1. Sequential decoding

The payload of each macroblock is ordered in raster scan order, if no tiling is present. So, no significant bitstream buffer is necessary in order to decode a macroblock in raster scan order, and there is no need to buffer the full decoded image.

4.3.1.10.2. ROI decode

Fast ROI decoding are allowed by using the tile partitioning of an image. Because each tile can be entropy coded independently, only that related to the ROI need to be decoded; the other tiles can be

ignored. This is useful in scenarios where only a small part of a stored large image needs to be rendered at any given instant.

4.3.1.10.3. Spatial scalability

JPEG XR offers full resolution image decoding and fast 16:1 and 4:1 thumbnail decoding ability. If a 16:1 thumbnail is required, only the DC subband needs to be decoded. A 4:1 thumbnail can be generated using only DC and LP subbands. 16: 1 thumbnailing does not require any inverse transform and 4:1 requires only the second stage transform. This feature is very important for zooming in and out of large images.

4.3.1.10.4. Quality scalability

It is allowed generating full resolution image with lower quality if some subbands are unavailable. Scalability allows for easy transcoding and enables progressive decoding: the image quality is enhanced with the increase of available subbands. This feature is advantageous for displaying images transmitted through slow channel.

5. Proposed Architecture

5.1 Client-server image browsing architecture

The proposed architecture is shown in Figure 5.1.

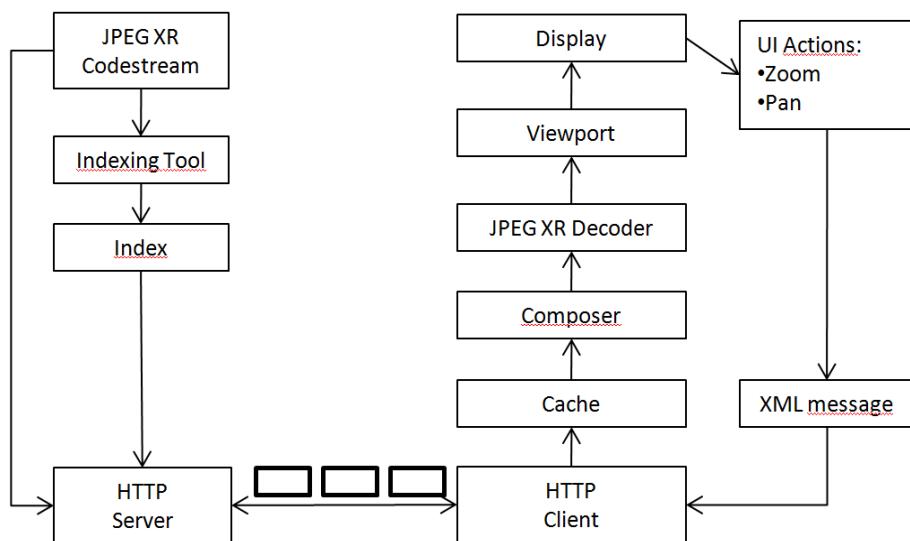


Figure 5.1 Reference Architecture [22].

It is a client-server architecture for remote browsing applications for the packets exchange over the Web. The server has been implemented in C++ language and it is responsible for access management, allocation and release of resources. It stores a database of JPEG XR images and for each connection to a client user data are transferred by the means of the HTTP protocol.

Table 5.1 shows the definition, both at server side and at client side, of an array that records the information already available to the client side.

An exchange protocol is also defined to enable the randomly access to the level of image requested by the client, as shown in Figure 5.2.

LIST	TILE #	LEVEL #	AVAILABLE
0	1	DC	[Y/N]
1	1	LP	[Y/N]
2	1	HP	[Y/N]
3	1	FLEX	[Y/N]
4	2	DC	[Y/N]
5	2	LP	[Y/N]
6	2	HP	[Y/N]
7	2	FLEX	[Y/N]
8	3	DC	[Y/N]
9	3	LP	[Y/N]
10	3	HP	[Y/N]
11	3	FLEX	[Y/N]
...

Table 5.1. Matrix of data available in the client side.

List 0 uses only DC coefficients to decode an image, List 1 uses DC+LP coefficients, List 2 considers DC+LP+HP coefficient, while in List 4 the decoding is a full resolution and so on.

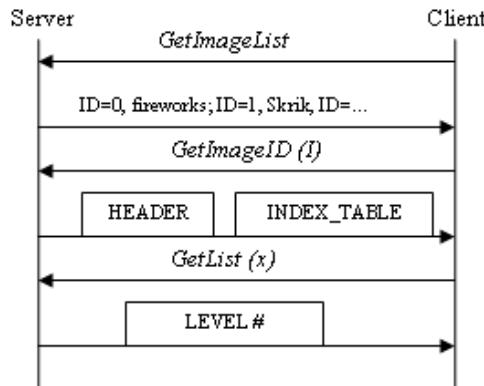


Figure 5.2. Client-server session [22].

The available commands are as follows:

- GetImageList, request for the list of images stored in the database;
- GetImageID(I), request for the HEADER and the INDEX_TABLE of the selected I image;
- GetList(x), submission of the tile having LIST x in the matrix

At the server side JPEG XR images are stored with an optimized tile decomposition [17], in order to allow fast local access by the client device. An indexing tools analyses the JPEG XR codestream in order to extract the offset and the size of each subband. Indexes are stored in the

index table for fast retrieval of the required information for the reconstruction of a ROI requested by the client. Index table is managed by the functional Index block in Fig. 8. This block has also the purpose of keeping track of the information already available at the client side and avoiding retransmission of the same data. The HTTP server receive incoming request and provide the chunks of data required by the HTTP client.

The client constructs the request on the basis of the current user view. In particular the display resolution and the level of zooms are used for determining what are the needed chunks of information and the corresponding indexes.

The proposed concept is slightly different from the concept used in the interactive protocol JPIP. In fact in the proposed architecture the computation of required chunk indexes is performed at client side, minimizing the information transferred for the request to a vector of index corresponding to the chunk number needed for display the current view to the user.

Moreover it is not needed to have an arbitrary ROI access but simplifying as much as possible the user interaction with the process of viewing high quality high resolution large images with mobile devices.

HTTP client receives incoming packet from the HTTP server containing JPEGXR image subband that are stored into the local cache. The block composer prepare the JPEG XR file merging the required chunks into a correct JPEG XR format in order to be decoded by the JPEG XR block.

The viewport block keep tracks of the visible portion of the image which is larger than the visualization device.

The user interacts with the display with classical image viewing operation such as zooming and panning. Since the information required are contained within the tile, the browsing speed increase with decreasing size of the tile. At the same time, however, the use of tiles of reduce size may cause a significant increase in terms of overhead.

5.2 Error recovery algorithm

Image transmission over error prone channel is a problem that has been deeply analyzed for previous JPEG standard, while researches for addressing this problem for the novel JPEG XR standard are currently in progress. JPEG XR does not provide any means for error recovery/concealment, hence the reference software is not able to decode an image affected by errors. This behaviour, in the specific case of video, may result in a significant deterioration of image quality and in many other cases the loss of the entire frame.

The reference decoder has been modified in order to add a very simple tool for error recovery based on the re-alignment of the VLC decoder position to the next macroblock after an error has been detected. To reach this aim, the expected starting position of each HP macroblock are saved as a separate header information.

The adjustment of JPEG XR decoder provide good quality even in presence of simulated bit errors in the HP band of the encoded image; the decoding process, at the same time, introduces a lower encoding overhead thanks to the macroblock index saving.

This is not sufficient to produce decode image of good quality, in presence of high levels of errors, because of the behavior of adaptive VLC coding and adaptive coefficient normalization mechanism. The adaptive coefficient normalization involves changes of the adaptive coding model: the more significant bit are VLC coded while the other variables are used to determine the number of least significant bits that form the FLEXBITS of an HP coefficient.

The adaptive model is initialized at the beginning of each tile and according to the statistics concerning the previous coefficients it is updated for each macroblock. If the evaluation of such coefficients is distorted because of errors that propagate to the next macroblock until the end of tile, the updating of the adaptive model can be wrong.

So the adaptive structure could be saved along with the starting position of each macroblock during the encoding and recovered during the decoding process. Because the presence of errors can also alter the VLC decoder context, the context will be reset at each macroblock, since it is initialized at the beginning of each tile..

The simplicity of the proposed algorithm adds very little computational overhead and seems very promising as confirmed by objective image quality results in experimental tests.

6. Experimental results

6.1 Region of interest (ROI) and tiling

Services for high definition image browsing on mobile devices require a careful design since the user experience is heavily depending on the network bandwidth, processing delay, display resolution, image quality. Modern applications require coding technologies providing tools for resolution and quality scalability, for accessing spatial regions of interest (ROI), for reducing the domain of the coding algorithm decomposing large images into tiles. This need occurs primarily in medical imaging applications, video surveillance systems, historical and GIS (Geographic Information System) images and in a considerable number of internet applications.

JPEG XR, such as JPEG 2000, allows the user to define region of interest (ROI) within an image that will be coded and transmitted in better quality and less distortion than the rest of the image. ROI processing is used if it is necessary to process a single sub-region of an image while leaving other region unchanged. The resulting ROI coefficients have a reduced number of transmitted bit-planes. An ROI can extend across multiple tiles and could not be aligned with tile boundaries. Entropy decoding process for a tile is adaptive and may also include a greater region of the image, causing a resource overhead which strongly depend on the image division into tiles, the tile sizes and the alignment between the tiles and ROIs.

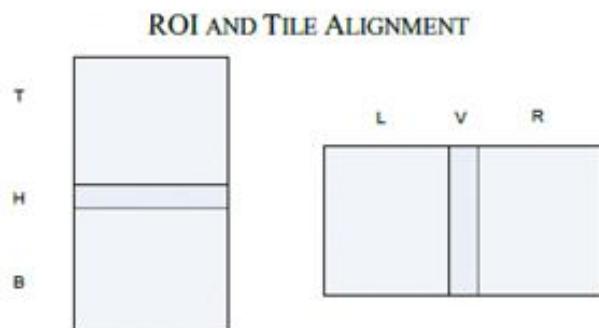


Figure 6.1 How extend ROIs overlap: left (T and B overlap by H horizontally), right (L and R overlap by V vertically) [15].

In the Figure 6.1, H represents the area of horizontal overlap between T and B tiles, while V represents the area of vertical overlap between L and R tiles. The decoding overhead depends on the location of H and V areas related to tile boundaries.

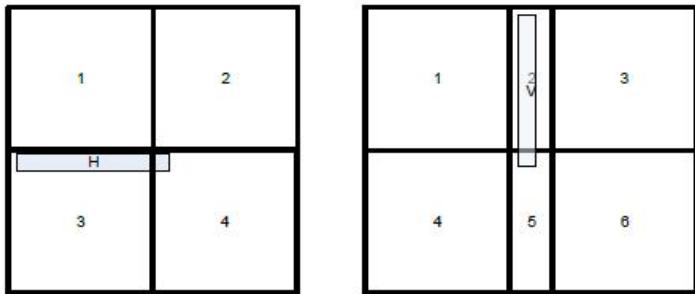


Figure 6.2 Optimal location of H (left) and V (right) [15].

Figure 6.2 shows the optimal position of H and V regions that minimize the overhead. In the first case, H is located slightly below the tile boundaries. If the H area extends between 1 and 2 tiles, in order to decode the B region is necessary to completely decode the two tiles. In the second case, the V region should be included, as much as possible, between the tile boundaries that are not very far from each other. Because the V region decoding implies the decoding of tile2, if tile 2 is not tall and slim, the decoding of L or B region or both will produce a substantial overhead.

The JPEG 2000 standard compress each tile as an independent image from the other tiles. It implies that decoding any region of interest (ROI) inside a tile requires only the coded codestream of that tile. The tile are processed through the wavelet transform, (that can be reversible (RCT) or irreversible (ICT), to obtain different subbands, that are respectively LL, HL, LH and HH.

The JPEG XR standard is slightly different. A source image is divided into a grid of macroblock-aligned tiles, that enable fast local access. JPEG XR encoder can choose among three overlap filter operation, as described above: 1) non-overlapping that implies that no overlapping operation is performed; 2) one-level overlapping filter; 3) two-level overlapping filter. It is also possible decide whether to handle the tile boundaries in soft mode or in hard mode. In the first case, the overlapping filter is enabled within the tile and over the edges between the tiles. In the second case, the overlapping operation is applied only within the tile. In order to reconstruct a specific ROI, if the adjacent macroblock is placed in a distinct tile, it is necessary decode only a part of that tile.

Moreover, a structure for an optimized tile construction exists[15] and uses 256x256 ROI to reduce around 1% the overhead required to decode an image.

In the optimized case we can find one of these possibilities:

- you want to decode a ROI that lies between two vertical tile;
- or you want to decode a ROI that lies between two horizontal tile.

In the first condition, the ROI should be on a small case separated from the edge of the horizontal tile. In this way it is not necessary decode the tile that are above.

In the second occurrence the ROI should be crushed into two neighbours vertical tile boundaries for the same reason of the previous case.

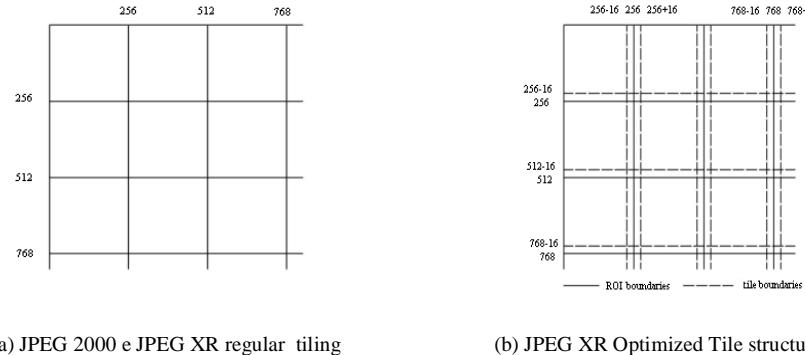


Figure 6.3 Regular tiling (a) and optimized tiling(b).

In JPEG XR image browsing applications, the optimized tile decomposition is necessary to reduce the amount of transferred data, especially when it comes to large images. Even with access to high-speed networks, it is in fact often impractical to transmit a large image as a single item. A simple solution is to maintain a low-resolution "thumbnail" images for each of the large images.

In the previous chapter an architecture for remote browsing of large image using fast local access means provided by JPEG XR has been proposed. A tile-based approach allows the user client to access to a desired ROI at a given resolution. JPEG XR coded file format should make use of the frequency mode order where the bitstream of each tile is set up as a hierarchy of bands. The main goal is to transmit only some subset of the available sub-band coefficients.

However, JPEG XR bit-stream is strongly vulnerable when transmitted over an error channel and unlike the JPEG 2000, the JPEG XR decoder, is not able to decode a corrupted file. This is due to the adaptive nature of the entropy encoding process of the JPEG XR code-stream. A single wrong bit within the bit-stream causes the wrong interpretation of a VLC coefficient and in most cases a completely mismatched decoding of the following coefficients, leading to a mistake in the

calculation of the next macroblock starting position in the bit-stream. The JPEG2000 partitions the code-stream into different segments (send first the most important information and then the details in a recursive manner) and this helps to isolate faults in a segment and to prevent their spread through the entire code-stream. Obviously an error on the details, however, allows reconstructing the image, because the loss is not important. JPEG XR delivers code-stream but with no significant distinction between info and details, so a bit of an error can significantly distort the picture completely.

6.2 Image scalability

This section contains the tests performed and their results. Tests were conducted using the JPEG XR reference software and the KAKADU software [32]. The use case is as follows: once the desired ROI at a given resolution has been selected, the client must be able to zoom in and out the image that appears in his own device, or make horizontal or vertical scrolls of the current view of the image.

It is possible to request only some subset of the available sub-band coefficients. In fact, in the frequency mode, each sub-band can independently be decoded. All tiles of a particular sub-band are merged together in a unique data packet.

This allows creating smaller image previews using a resolution that fits the device used to load the image.

The client can request only the DC level transmission, if a low quality of image suffice. It is possible to increase the image quality, transmitting progressively all available sub-bands.

DC+LP+HP+FLEX sub-bands transmission ensures maximum image quality.

Removing irrelevant information obviously causes a loss of data image that must be quantified in some way. There are subjective and objective techniques. The subjective metric depends on the experience of the observers, while the objective metric consists to calculate the difference in statistical distribution of pixel values in digital images; in this way it is possible to quantify the distortion of the compressed image with respect to the original. The PSNR is the most widely used image objective metric.

The bitrate vs PSNR shows the efficiency of a compression algorithm: high bitrates provide a higher quality of the compressed data.

Several tests were performed using the three different overlapping filters and distinguishing between hard, soft and optimized mode. These results are then compared with those obtained with JPEG 2000 tests.

Image request at difference resolution level have been analyzed in order to report the bit-rate of the data transferred at each requests.

The following figures show the experimental results. Each figure contains only the average of the results of the most significant images used as a test set.

Experimental tests were carried out by defining three mode of encoding JPEG XR (soft, hard and optimized) and using the three OT mode (0,1,2). Subbands considered are: 1) only DC, 2) DC+LP, 3) DC+LP+HP, 4) DC+LP+HP+FLEX.

JPEG2000 codestream were produced in relation to JPEG XR bitrate, to make a comparison between the two coding algorithms.

In the first JPEG XR case (Fig. 6.4(a-b-c)) the solution without any level of overlap was considered; in the second JPEG XR case (Fig. 6.5 (a-b-c)) only one overlapping filter was applied, while in the latter JPEG XR case (Fig. 6.6 (a-b-c)) both stages of overlapping have been taken into account. In all three cases there was a distinction between soft (a), hard (b) and optimized (c) mode, in order to compare this three methods.

As expected, experimental tests, in terms of bpp, show that in the case with $L = 0$ hard, soft and optimized mode show similar results. In the case with $L= 1$ soft and optimized mode show similar results, while the bpp slightly decreases in hard mode.

Finally, in the case with $L= 2$ optimized mode performance in terms of bpp presents results that are intermediate between those obtained with hard and soft mode.

In all experimental JPEG XR tests, therefore, hard tiles outperform soft tiles and optimized tile; but the results obtained with the tile optimization, in terms of overhead, are very acceptable if compared with those of JPEG 2000.

Fig. 6.7 shows the JPEG 2000 results obtained in the case of image subdivision in tiles of 256x256 size, without any type of optimization.

It is possible to notice that the JPEG 2000 bpp are very similar to those obtained with JPEG XR experiments, but in JPEG2000 we can see a gradual increase of bpp as bitrate increased, even if only DC.

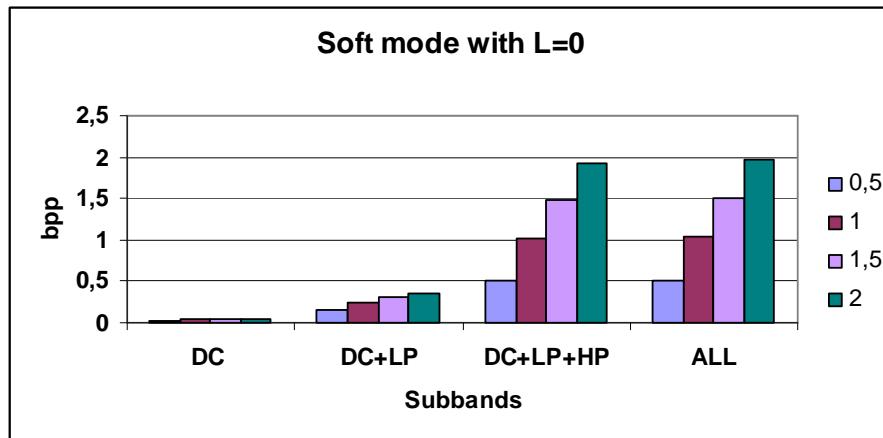


Figure 6.4 (a). Bitrate for each JPEG XR subband for four target bitrates (soft mode, L=0).

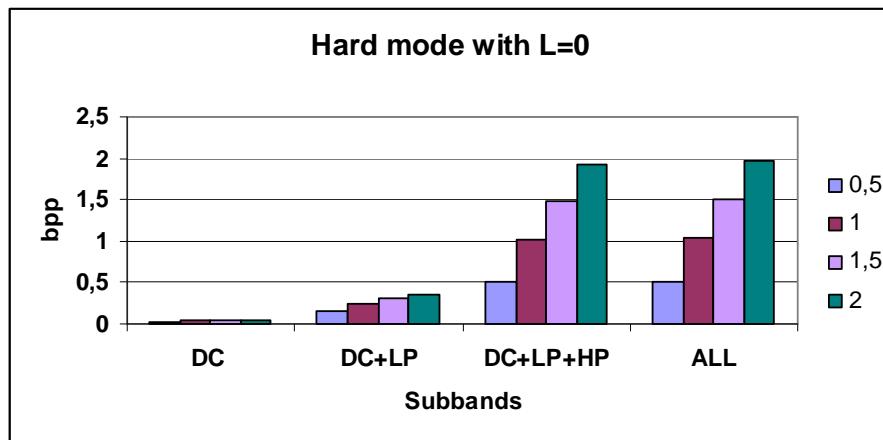


Figure 6.4 (b). Bitrate for each JPEG XR subband for four target bitrates (hard mode, L=0).

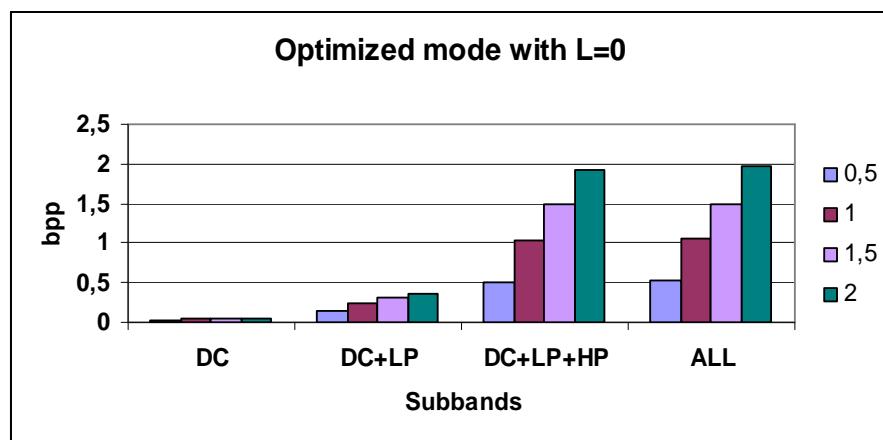


Figure 6.4 (c). Bitrate for each JPEG XR subband for four target bitrates (optimized mode, L=0).

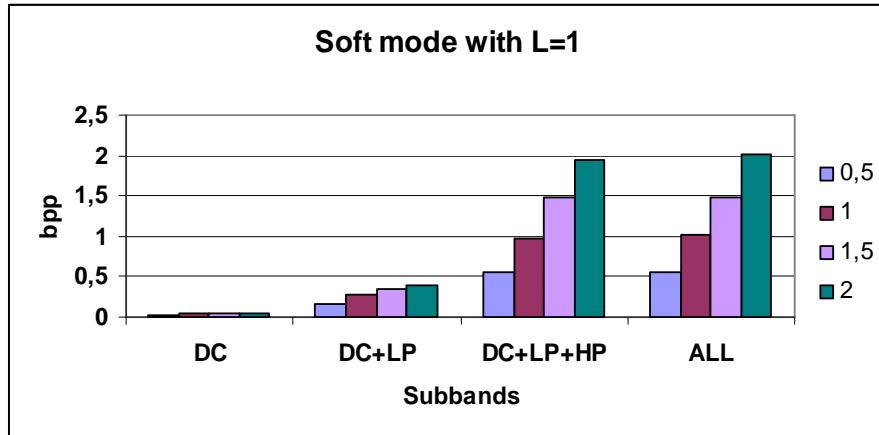


Figure 6.5 (a). Bitrate for each JPEG XR subband for four target bitrates (soft mode, L=1).

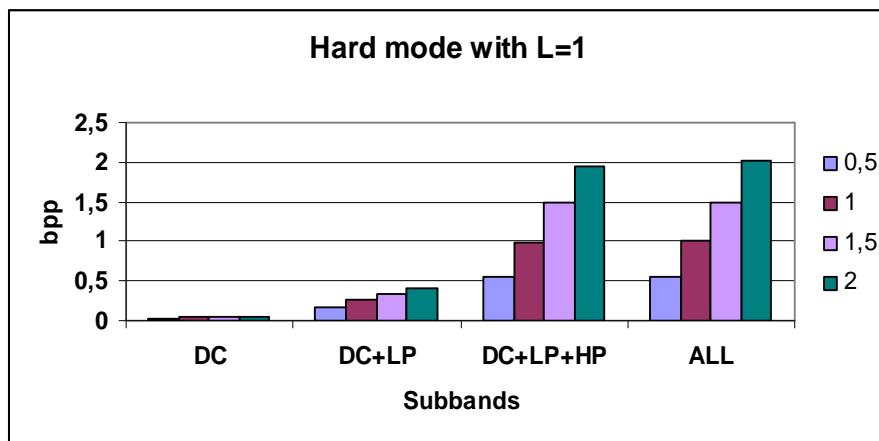


Figure 6.5 (b). Bitrate for each JPEG XR subband for four target bitrates (hard mode, L=1).

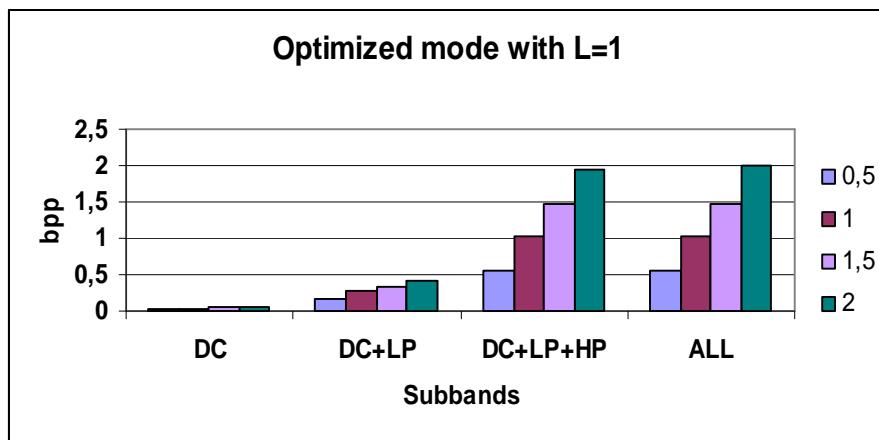


Figure 6.5 (c). Bitrate for each JPEG XR subband for four target bitrates (optimized mode, L=1).

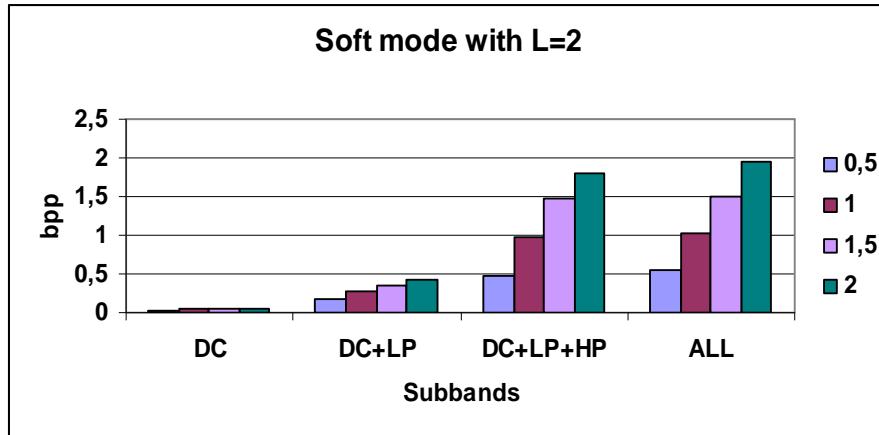


Figure 6.6 (a). Bitrate for each JPEG XR subband for four target bitrates (soft mode, L=2).

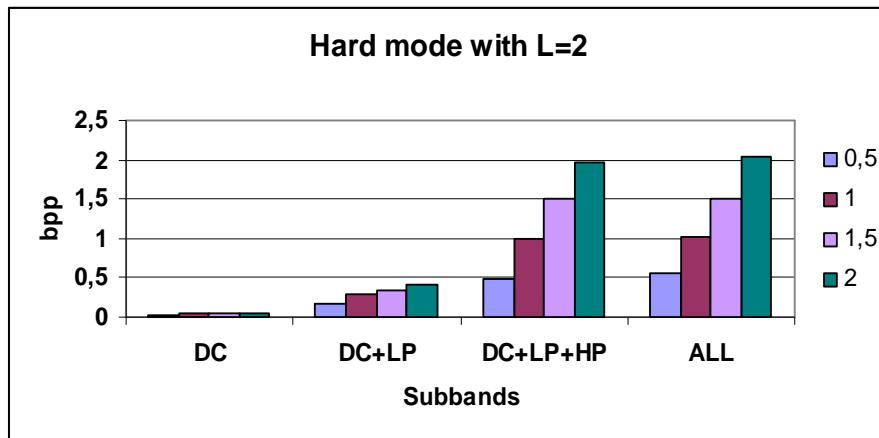


Figure 6.6 (b). Bitrate for each JPEG XR subband for four target bitrates (hard mode, L=2).

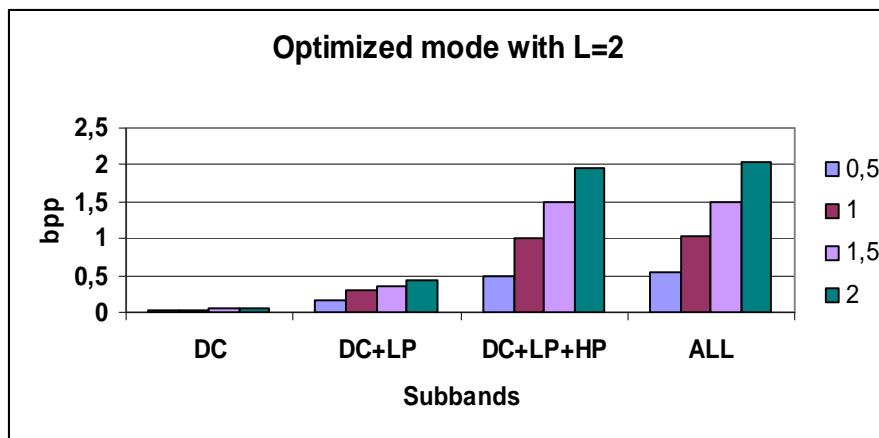


Figure 6.6 (c). Bitrate for each JPEG XR subband for four target bitrates (optimized mode, L=2).

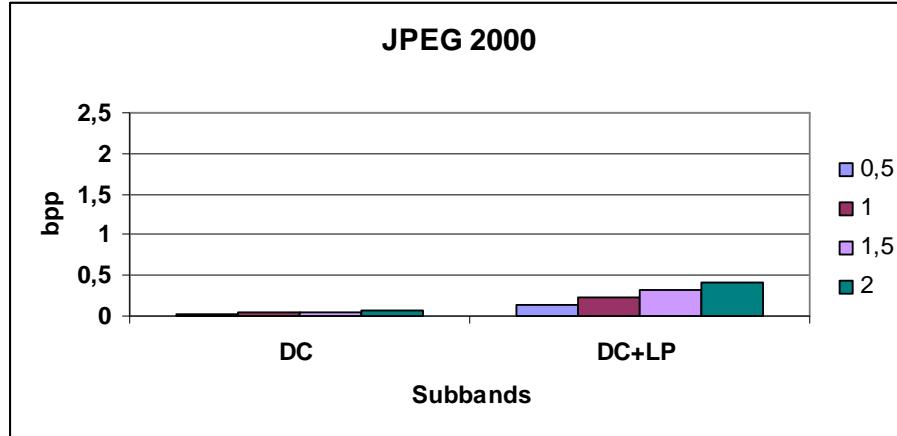


Figure 6.7 Bitrate for each JPEG 2000 subband for four target bitrates.

6.3 Transmission over error channels

In order to analyze the JPEG XR coding robustness in transmission over error-prone channels, first, it was simulated the transmission of an encoded image on a symmetric binary channel, which produces the randomly change of some characters in the picture, according to the probability of error. Three error probabilities 10^{-2} , 10^{-3} and 10^{-4} have been considered and the tests were performed on a set of four images.

JPEG, JPEG 2000 and JPEG XR standard encoding were compared to verify the effectiveness of their performance in real time applications. Some metrics usually used to evaluate the effectiveness and the efficiency of image coding have been adopted. In particular, the comparison of performance is achieved in terms of options and features based on the codec, image quality and PSNR. These metrics are also the main demand of digital photography, it has to be guaranteed a good compromise between encoding options, quality, use of memory and easy of embedded coding. The derived results of experiments are summarize in Table 6.1.

Source image	Error probability	PSNR [dB] JPEG XR	PSNR [dB] JPEG 2000	PSNR [dB] JPEG
Bike	10^{-2}	4.40	6.70	6.06
	10^{-3}	5.00	23.00	4.99
	10^{-4}	8.00	27.29	8.41
Baboon	10^{-2}	5.20	7.20	7.08
	10^{-3}	5.90	26.68	8.56
	10^{-4}	6.50	30.28	10.38
Peppers	10^{-2}	5.36	7.18	9.08
	10^{-3}	4.62	28.84	4.20
	10^{-4}	6.14	35.97	17.60
Woman	10^{-2}	5.31	6.40	7.39
	10^{-3}	6.17	32.00	6.04
	10^{-4}	6.25	34.17	8.95

Table 6.1 Experimental results

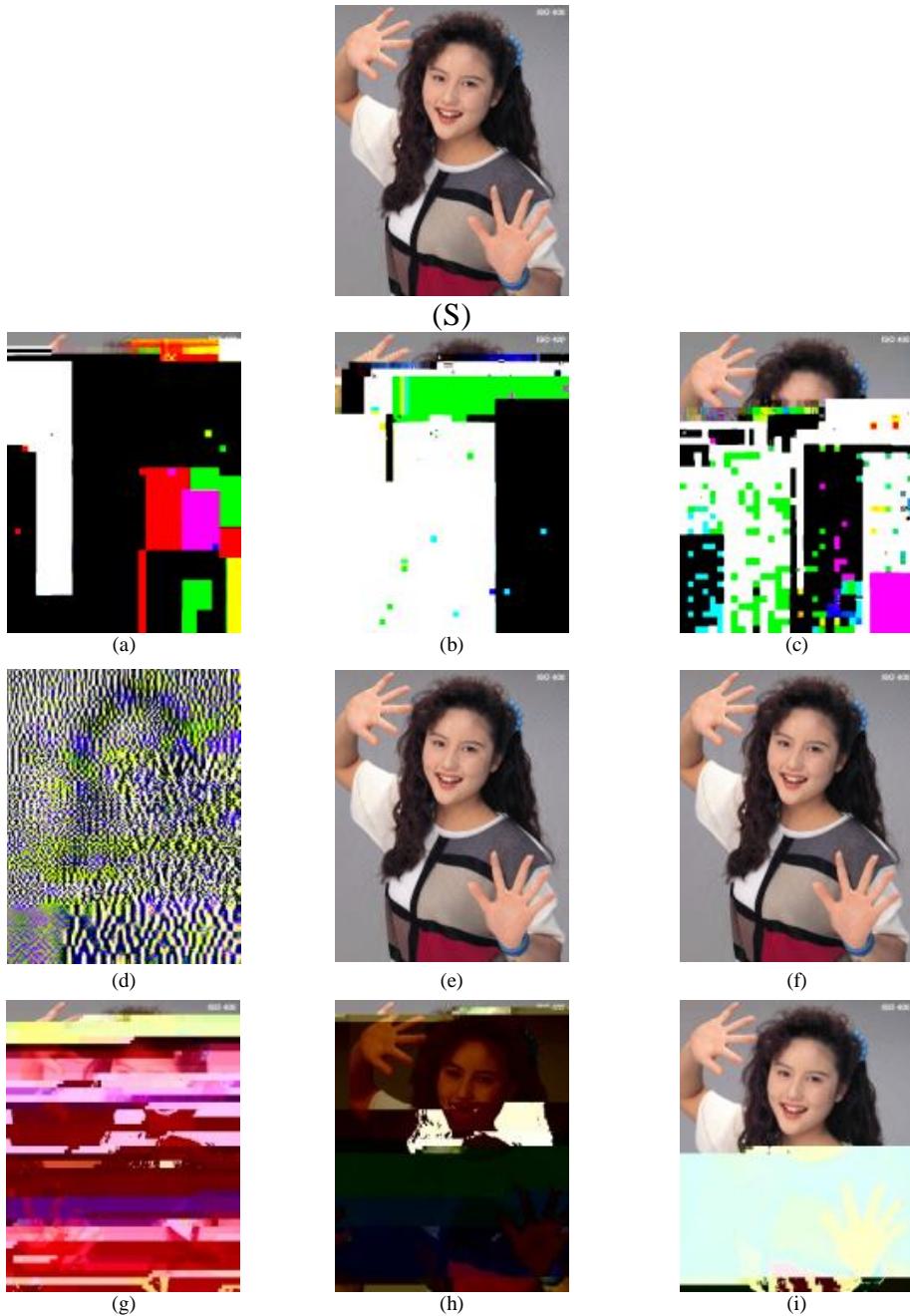


Figure 6.8 Visual results. Source image (S) , JPEG XR images for $10^{-2}, 10^{-3}$ and 10^{-4} probabilities (a,b,c), JPEG 2000 images for $10^{-2}, 10^{-3}$ and 10^{-4} probabilities (d,e,f), JPEG images for $10^{-2}, 10^{-3}$ and 10^{-4} probabilities (g,h,i).

Figure 6.8 shows a visual example.

It is possible to notice that, in terms of PSNR, at the same compression rate, JPEG 2000 provides means for error recovery, while JPEG XR is not robust to the transmission over a channel with error, presenting a worse performance compared to those of JPEG encoding.

The results of those tests shown therefore that JPEG XR is not able to recover error in data transmission.

To improve the JPEG XR image behavior a decoder modification has been done, in order to allow a re-alignment of the decoder reading position from a bitstream, using a different HP subband header information.

It was considered the JPEG XR approach in frequency mode, in which the bit-stream is divided in subbands with the following order; DC, LP, HP and FLEXBITS. The last sub-band can be absent, so it was not considered in our experimental tests.

Now, a test set of 50 source images encoded with the new JPEG XR decoder were examined. Random errors have been applied to the HP subbands in order to test the distorted image decoded.

Experimental tests were performed using different values of quantization coefficients (Q equal to 12, 20, 54 and 68), error probabilities (10^{-2} , 10^{-3} , 10^{-4} and 10^{-5}) and three options of overlapping filter (0, 1 and 2); those three elements are completely jointed. This means that a change of one of those parameters causes different image coded behavior.

To compare the results the PSNR metric has been utilized. It is possible to evaluate the image quality as a objective metric.

In particular it was evinced that for high error probabilities, coding images without any overlapping filter is less corrupted.

For each value of probabilities and overlap filter, the quantization parameter in the encoding process has been changed.

At low error probabilities it is very difficult to notice substantial difference among the images.

For high probabilities, instead, it is possible to note that better performance are obtained for corrupted encoded images without the application of any overlap level.

In general, experimental tests, for high values of quantization parameters shown that bit errors in the HP band of a quantized image are less significant.

Figure 6.9 shows the experimental results in terms of PSNR for different values of quantization coefficients and different probability.

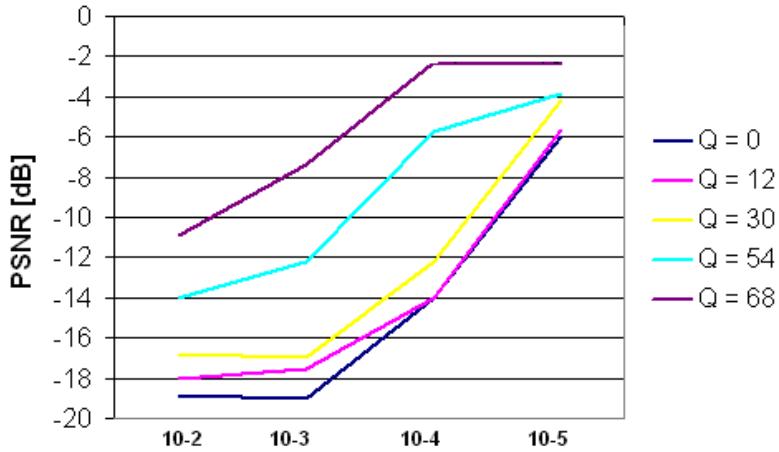


Figure 6.9 Average PSNR [dB] for corrupted images varying q values and probability error in the HP band.

Finally, the computational complexity has been evaluated, both in encoding and decoding process.

The encoding and decoding time was evaluated on a Pentium 4 Processor, 2.2GHz-M, 512MByteDDR SDRAM, and Microsoft Windows XP Operating System.

The following histogram in Fig.6.10 shows the obtained result for JPEG XR image decoder, considering a test set of four images. There is a comparison between the decoding time of JPEG XR without overlapping, and the decoding time with two level overlap filter both for JPEG XR decoder with OT=2 in the original version and in the modified one.

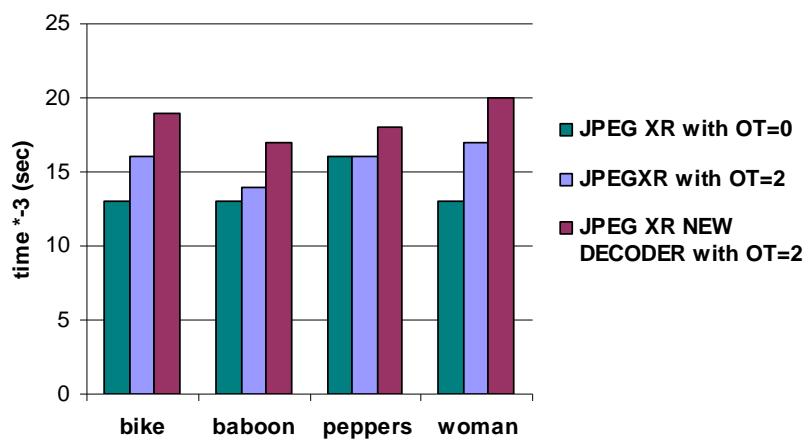


Figure 6.10 Histogram for computational complexity in decoding.

Experimental results indicates that the JPEG XR error recovery decoded provide an increase in overhead quite irrelevant in relation to the improvement of image quality.

7. Conclusions

The wide variety of devices for mobile imaging (PDA, Mobile Phone, and Smartphone) requires that the compressed image is capable to adapt to different viewing conditions.

The ISO/IEC JPEG [1] is universally accepted as the standard “de facto” in the field of lossy compression, despite it is very dated. JPEG is a compression algorithm based on DCT transform. Every image, after decomposition in 8x8 blocks, is sent to the encoder, which in turns consists of three simple steps: a FDCT, that converts the 8x8 block from the spatial to frequency domain, a quantizer that eliminates less important information and an entropy encoder to reduce the amount of images needed.

The JPEG 2000 standard [2] uses a different transform algorithm based on wavelet and an innovative system for entropy coding named EBCOT that provides better compression performance than JPEG at high compression ratios. Image compression both binary and continuous-tone, random access to the bit-stream, processing domain encoded and robustness to transmission errors are some of JPEG 2000 tools. Nevertheless, the new algorithm has not been as successful as hoped, despite all the novel functionalities.

The search for efficient representation for digital photographs is always alive and the ISO/IEC committee has recently completed the standardization process of the new JPEG XR image compression algorithm.

The JPEG-XR image standard [4][5], as well as JPEG 2000, has been developed to reduce the limitations of JPEG standard. JPEG XR algorithm was designed and implemented to achieve high performance in a transparent way to the user, using simple procedures, not differentiating between lossless and lossy compression, and allowing the management of a wide range of native formats. It allows representing multiple images into a single file, to decode only a part of an image, and crop, reduce in quality, tilt or rotate without having to decode the file. It adopts a method for compression image that allows HDR (High Dynamic Range) encoding, while using integers (without division) for both compression and decompression. This aspect allows providing low computational complexity.

In this thesis, an architecture for remote browsing of large images coded using fast local access means provided by JPEG XR has been presented. In order to minimize the transferred

information, the JPEG XR coded file format should make use of the frequency mode order and partitioning of images into tiles. The main goal is transmitting only some subset of the available sub-band coefficients at a given resolution.

This is necessary to allow an interactive access to portion of images (ROI), that are downloaded and displayed, minimizing the amount of data transferred and maintaining an acceptable image quality for the user experience. Interactive image browsing over the internet of tiled JPEG XR images provides user experience comparable to image browsing based on JPEG 2000.

The proposed architecture has of course prompted a study of errors in transmission on unreliable channel, such as the wireless one, and the definition of possible optimizations/variants of the codec in order to overcome its own limitations. So, an extension to the JPEG XR algorithm has been proposed; this consists in an error recovery process that enables the decoder to realign itself to the right bitstream position and to correctly decode the most part of the image.

Preliminary exploration of additional coding algorithms enabling JPEG XR error recovery has been presented. Error probabilities and QP values have been jointly investigated and experimental analysis has been done.

The obtained results have shown that JPEG and JPEG XR are less robust than JPEG 2000, in particular at very low bit error rates. JPEG XR does not provide tools for error recovery and further investigations is ongoing in order to define techniques for error detection and concealment of the corrupted bitstream.

Bibliography

- [1] G.K.Wallace, “The JPEG still picture compression standard,” Communications of the ACM, vol.34, no. 4, pp. 31-44, Apr. 1991
- [2] Skodras, C. Christopoulos, T. Ebrahimi, “The JPEG 2000 still image compression standard,” Signal Processing Magazine, IEEE vol. 18, Issue 5, 36-578 (September 2001).
- [3] E. Edwards and S. Foessel, “JPEG2000 for digital cinema applications”, <http://www.jpeg.org/public/DCINEMA-v2.pdf>.
- [4] ITU-T Rec. T.832 | ISO/IEC 29199-2, “Information technology – JPEG-XR image coding system – Image coding specification,” available online at: <http://www.itu.int/rec/T-REC-T.832>.
- [5] F. Dufaux, G. J. Sullivan and T. Ebrahimi, “The JPEG-XR image coding standard,” IEEE Signal Processing Magazine, Volume 26, Issue 6, November 2009, pp:195-199, 204-204.
- [6] ISO/IEC 29199-1: Information technology – JPEG XR image coding system – Part 1: System architecture, 2010
- [7] ISO/IEC 29199-2: Information technology – JPEG XR image coding system – Part 2: Image coding specification, 2010
- [8] ISO/IEC 29199-3: Information technology – JPEG XR image coding system – Part 3: Motion JPEG XR, 2010
- [9] ISO/IEC 29199-4: Information technology – JPEG XR image coding system – Part 4: Conformance Testing, 2010
- [10] ISO/IEC 29199-5: Information technology – JPEG XR image coding system – Part 5: Reference software, 2010
- [11] Ching-Yen Chien, Sheng-Chieh Huang, Chia-Ho Pan and Liang Gee Chen, “Full HD JPEG XR Encoder Design for Digital Photography Applications”, In Tech, Feb.2010.
- [12] Sridhar Srinivasan, Chengjie Tu, Shankar L. Reguanthan, and Gary J.Sullivan, “HD Photo: a new image coding technology for digital photography”, Proc. SPIE 6696, Applications of Digital Image Processing XXX Andrew G.Tescher, San Diego CA, Aug. 2007.

- [13] Lang Yu, Master thesis “Evaluation and Implementing JPEG XR Optimized for Video Surveillance”
- [14] Jie Liang Engineering Science Simon Fraser University, “ENSC 861- Source Coding in Digital Communications – Practical VLC Coding”, Jan.2011
- [15] Chengjie Tu, Gary J. Sullivan, and Sridhar Srinivasan, Microsoft Corporation, Redmond, USA “Optimizing JPEG XR tile structure for fast local access”
- [16] H. S. Malvar, “Efficient signal coding with hierarchical lapped transforms,” Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, Albuquerque, Apr. 1990, pp. 1519–1522.
- [17] C. Tu, G. J. Sullivan, and S. Srinivasan, “Effect of tile size on JPEG XR compression efficiency”, ISO/IEC JTC 1/SC 29/WG 1 N4980, Japan, April 2009.
- [18] Koichi Hattori, Hiroshi Tsutsui, Hiroyuki Ochi, and Yukihiro Nakamura, “A High-Throughput Pipelined Architecture for JPEG-XR Encoding,” Embedded Systems for Real-Time Multimedia, 2009. ESTIMedia 2009. IEEE/ACM/IFIP 7th Workshop on.
- [19] Jain and S. Panchanathan, “Scalable Compression for Image Browsing,” Consumer Electronics IEEE, vol.40, Issue3, August 1994.
- [20] Z. Pecenovic, M. Do, M. Vetterli, and P. Pu, “Integrated Browsing and Searching of Large Image Collections,” in International Conference on Advances in Visual Information Systems, 2000, pp. 279–289.
- [21] Jin Li, Hong-Hui Sun, “On interactive browsing of large images,” IEEE Transactions on Multimedia, vol. 5, no. 4, pp. 581-590, Dec 2003.
- [22] C. Perra, D.D. Giusto, “An image browsing application based on JPEG-XR,” International Workshop on Content-Based Multimedia Indexing, 2008, 18-20 June 2008, pp. 396-401.
- [23] L. Atzori, D.D. Giusto, A. Raccis, “High-frequency error concealment in wavelet-based image coding”, Electronics Letters, Vol.39, Issue 2, 23 Jan 2003, pp. 203-205.
- [24] C. Perra, P.A. Pes, D.D. Giusto, "High-frequency error recovery in JPEG XR coded images," IEEE International Conference on Image Processing , 2011, 11-14 September 2011.
- [25] C. Tu. S. Srinivasan, G.J. Sullivan, S. L. Regunathan and H. S. Malvar, “Low-complexity hierarchical lapped transform for lossyto lossless image coding in JPEG XR / HD Photo”, SPIE Appl. Dig. Image Proc. XXXI, vol 7073, paper 7073-12, Aug. 2008.
- [26] X.Fan, X.Xie, W.Y. Ma, H.J. Zhang, H.Q. Zhou, ”Visual attention based image browsing on mobile devices”, proceedings of ICME, vol.I, pp. 53-56, Baltimore, MD, 2003

- [27] Chen L.Q., Xie X., et al., "A visual attention model for adapting images on small displays", ACM Multimedia Systems Journal, vol. 9, no. 4, p353-364, Oct. 2003.
- [28] Xing Xie, Hao Liu, Wei-Ying Ma, Hong-Jiang Zhang, "Browsing large pictures under limited display sizes," IEEE Transactions on Multimedia, vol. 8, no. 4, pp. 707-715, Aug. 2006
- [29] Hasan, M.A., Changick Kim, "An Automatic Image Browsing Technique for Small Display Users", Proceeding ICACT'09 Proceedings of the 11th international conference on Advanced Communication Technology - Volume 3
- [30] Xin Fan, Xing Xie,Wei-Ying Ma, "Detecting the sufficient display resolution for image browsing", mdm, pp.22, 7th International Conference on Mobile Data Management (MDM'06), 2006
- [31] H.Liu, S.Jiang, Q.Huang, Xu, W. Gao "Region-based visual attention analysis with its application in image browsing on small displays", In proceedings of the 15th international conference on Multimedia, pp.305-308,2007.
- [32] Kakadu Software Home, [Online], Available: <http://www.kakadusoftware.com>
- [33] De Simone Francesca, Ouaret, Mourad Dufaux, Frederic, Tescher Andrew G., Ebrahimi Touradj, "A comparative study of JPEG 2000, AVC/H.264, and HD Photo", SPIE Optics and Photonics, Applications of Digital Image Processing XXX, San Diego, CA USA, 28 - 30 August 2007
- [34] Mourad Ouaret, Frederic Dufaux, Touradj Ebrahimi, "On comparing image and video compression algorithms", Third International Workshop on Video Processing and Quality Metrics for Consumer Electronics VPQM-07, Scottsdale, Arizona, USA, January 25-26, 2007.

Related Papers

- **B. Saba, C. Perra, D.D. Giusto,** "Interactive image viewing in mobile devices based on JPEG XR", Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, 2012, ISSN : (in press)
- **B. Saba, C. Perra, D.D. Giusto,** "Image transmission over error-prone channels", 19th Telecommunications forum (TELFOR), 2011, ISBN: 978-1-4577-1498-6 (Relation to Chapter 5 and 6)
- **B. Saba, C. Perra, D.D. Giusto,** "Interactive image viewing in mobile devices based on JPEG XR," 7th International ICST Mobile Multimedia Communications Conference (MOBIMEDIA), 5-7 September 2011. (Relation to Chapter 5 and 6)