Departament de Teoria del Senyal i Comunicacions
Universitat Politècnica de Catalunya

Tesi Doctoral

Compressió d'Imatge basada en la Codificació de Formes Perceptuals

(Image Compression based on
Perceptual Coding Techniques)

Autor: Josep R. Casas Pla
Director: Luis Torres Urgell

Barcelona, Març de 1996

A l'Àngels

Que la força del Sol, l'enginy de l'aigua i la sonoritat del vent
ajudin aquest pobre vell trobador
per tal d'explicar a la Humanitat sencera
la història del jove Arís, fill de Roger i Garidaina,
al Jardí de les Palmeres, en cerca del Secret de l'Aigua.
I que els nobles lectors, presents i futurs
sàpiguen perdonar la feblesa d'una memòria,
estovada pels anys i encegada per l'Amor. Així sia.

El Jardí de les Palmeres
JAUME FUSTER

Tesi Doctoral:

COMPRESSIÓ D'IMATGE BASADA EN LA CODIFICACIÓ DE FORMES PERCEPTUALS

Autor: Josep R. Casas Pla
Director: Luis Torres Urgell

RESUM DE LA TESI

En aquesta tesi s'estudien els mètodes de codificació d'imatges i seqüències de video des del punt de vista de la forma en què el sistema visual humà percep i entén la informació visual. La relevància d'aquest estudi ve donada pel paper tan important que tenen els senyals d'imatge en la civilització actual i pel gran volum de dades que representen les fonts d'informació visual pels sistemes que les han de processar.

S'han estudiat tres aproximacions per a la codificació de textures en un esquema avançat de compressió fonamentat en aspectes de percepció visual. La primera aproximació es basa en les transicions de la imatge i estudia la interpolació d'àrees suaus a partir de les esmentades transicions. La segona contempla l' extracció, selecció i codificació de detalls significatius per al sistema visual humà. Finalment, la tercera aproximació estudia la representació eficient de les textures fines i homogènies, que donen una apariència natural a les imatges sintetitzades aconseguint elevades tasses de compressió. Per a l'aplicació d'aquestes tècniques a la codificació d'imatge i video, es proposa un model d'imatge de tres components adaptat a les característiques perceptuals de la visió humana.

Les aproximacions de codificació objecte de l'estudi han portat al disseny de tècniques noves d'análisi i codificació d'imatge. A partir d'eines no lineals de tractament obtingudes de l'entorn de la Morfologia Matemàtica, s'han desenvolupat tres tècniques de codificació de textures. En concret,

- Un mètode d'interpolació "morfològica" orientat a la resolució del problema d'interpolació de senyals bidimensionals a partir de conjunts arbitraris de punts dispersos.

- S'ha introduït de manera experimental un criteri subjectiu empíric per a la ordenació i selecció de detalls en les imatges, segons un criteri perceptual.

- Finalment, s'ha investigat l'aplicació d'una tècnica clàssica, la codificació "subbanda", a l'interior de regions de forma arbitrària, resultant en un nou mètode de codificació de textures anomenat "Region-based subband coding".

Aquestes tècniques han estat novedoses en el camp de codificació d'imatge entre les anomenades tècniques orientades a objectes o de Segona Generació. Tanmateix, el model d'imatge estudiat, es troba en la línia de les últimes propostes en l'entorn de l'MPEG4, el futur standard per a comunicació d'imatge a baixa velocitat, que contempla la possibilitat de la manipulació de continguts.

Doctorate Thesis:

IMAGE COMPRESSION BASED ON PERCEPTUAL CODING TECHNIQUES

Author: Josep R. Casas Pla
Advisor: Luis Torres Urgell

SUMMARY

This thesis studies image and video sequence coding methods from the point of view of the way the human visual system perceives and understands visual information. The relevance of such study is due, on the one hand, to the important role that visual signals have in our civilization and, on the other hand, to the problem of representing the large amount of data that image and video processing systems have to deal with.

Three different approaches have been investigated for the coding of image textures in an advanced compression scheme relying in aspects of visual perception. The first approach is based on image transitions and the interpolation of smooth areas from such transitions. The second one, considers the extraction, selection and coding of meaningful image details. Finally, the third approach studies the efficient representation of homogeneous fine textures that give a natural appearance to the reconstructed images at high compression levels. In order to apply these techniques for still image and video coding, a three component model of the image, that matches the perceptual properties of the human vision, is put forward.

The coding approaches subject of research have leaded to the design of three new image analysis and coding techniques. Using non-linear tools from the framework of Mathematical Morphology, three texture coding techniques are developed. In particular,

- A "morphological" image interpolation method aimed at the problem of scattered data interpolation.
- An empirical subjective criterion for the ranking and selection of image details according to visual perception.
- The application of a conventional image coding technique, subband coding, to the coding of arbitrarily shaped image regions (region-based subband coding).

These are new texture coding techniques in the field of object-oriented and Second Generation image and video coding schemes. Furthermore, the model of the image that has been investigated follows the line of the last proposals in the framework of MPEG4, the forthcoming coding standard for low bit-rate visual communications, which considers the possibility of content-based manipulation and coding of visual information.

As a bandwidth compression device, the retina extracts visual information which is important to us for survival; its job is to extract just the information necessary and some discriminations to be made about objects in the outside world [...]. It is worth knowing that in animals which are prey rather than predators, many of these discriminations are made in the retina layer. This enables such animals to react more quickly to sudden attacks by their predators, and their retinas exhibit a great deal more neural interconnections than ours as a result. As a predator, man has had the luxury of being able to think about his visual world.

[...]

Nature has evolved our visual system in a long series of trials and failures spanning many millions of years. The laws of physics governing image formation and detection have presumably been the same over this span of time, and the evolution of vision had to work under these laws just as the image processing engineer does today. By taking this view we should not be surprised that an image-processing algorithm based on human vision often will provide a good physical solution to the problem as well. Rather, we should expect it to.

The Role of Human Visual Models in Image Processing
Douglas J. Granrath

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The ever tighter weave of communications, computing, networking and entertainment services has assumed a dominant role in our everyday life. The increasing availability of personal workstations and advanced communication channels fosters the vision of a world in which any kind of information flows freely among a variety of systems. However, despite the rapid progress in mass-storage density and digital communication systems performance, demand for data transmission bandwidth and storage capacity continue to outstrip the capabilities of available technologies. The growth of data-intensive digital audio and video applications and the increasing use of bandwidth-limited media such as radio and satellite links have not only sustained the need for more efficient ways to encode these signals, but have made signal compression central to digital communication and signal-storage technology.

In particular, the important role that image signals play in our civilization is being transferred to this new world of information technologies. Digital image and video applications require high transmission rates, large storage capacities and fast processing equipment, if the image data is handled in its raw form. Typical television images, for instance, generate data rates exceeding 100 Mbit/s. The emergence of new visual communication systems poses the problem of how to compress such a huge amount of information into a limited bit-rate for transmission or storage purposes. Examples include communication systems ranging from bit-rates about 20 Mbit/s to rates below 64 kbit/s; from high definition television, aimed at the highest visual quality [30], to very low bit-rate applications [58] such as video-phones, mobile image communications, electronic newspapers, surveillance systems or communication aids for the deaf, where it is impossible to reach the target compression while still keeping high quality of the decoded images.

International standardization efforts related to the different application areas have been made, resulting in a number of well-known standards: MPEG-2 [112] for digital television broadcasting at rates above 2 Mbit/s, MPEG-1 [56] for digital video rates of 1.5 Mbit/s and H.261 [61] or the forthcoming standard H.263 [42] for low and very-low bit-rate (less than 64 kbit/s) services related to visual telephony. In addition, the flexibility of new communication systems and the success of multimedia databases require interactive capabilities for content-based access and manipulation of audio-visual information. Besides compression, new content-based *functionalities* are currently under research within the framework of the future MPEG-4 coding standard [70].

1.1 The problem of image compression

The problem of compression is often referred to as *low bit-rate coding* or *coding* for short. The primary design objective in image compression is to minimize the average number of bits used to represent a given image or video sequence in digital form. Digital image compression techniques have made impressive progress in recent years. In the most demanding very low bit-rate applications, compression systems seem to be confronted with the awesome challenge of ‘getting something for nothing’. For example, a compression ratio of 50:1 simply means that 98% of the original data in the image has been eliminated.

Compression can be achieved by removing *unnecessary* information about the images. However, the only type of information that can be removed without noticing any degradation in image quality are:

- redundant information, which can be accurately predicted
- information that the human visual system cannot perceive.

1.1.1 Lossless and lossy image coding techniques

Lossless compression techniques aim at the exact reconstruction of the original images. This can be done if only redundant information is discarded. *Redundancy* is a characteristic related to factors such as predictability, randomness and smoothness of image data. Due to statistical properties of spatial distributions of luminance and color signals in natural images, little information can be predicted without distortion. Therefore, lossless compression systems seldom reach significant values of compression.

Strictly speaking, lossless image coding techniques should not allow any loss in signal to noise ratio (SNR). Nevertheless, if an average viewer cannot detect any difference when the original and the reconstructed compressed images are seen under normal viewing conditions, the compression system is said to be *visually lossless* [38]. Visually lossless compression can only be done by leaving out unnecessary information, i.e., either redundant information or information that the human visual system cannot perceive.

As we seek lower bit-rates in the digital representation of images, it is imperative to design the compression algorithm both to reduce redundancy in the input image and to remove the least relevant information from a perceptual point of view. Among *lossy compression* techniques, those which put special emphasis on the second operation are the most appropriate for very low bit-rate coding applications. When quality losses cannot be avoided in the reconstructed images, the compression system should do its best in order to make such losses hardly perceptible.

In this thesis, we have investigated several image and video coding techniques which take into account the perceptual point of view. The particular contributions in this field are summarized at the end of this chapter. Let us first present an overview of the main concepts involved in *perceptual coding* that have led us to the study of such techniques.

1.1.2 Perceptual coding

Central to the idea of visually lossless compression is the notion of *distortion masking.* When the distortion introduced in the coding process is properly distributed, it can be masked by the original image content. Perceptibility of distortion can be zero even if the objectively measured local SNR is modest or low. Ideally, if the noise level at all pixels in the image is exactly at the level of the *just noticeable distortion* (JND), the compression system would yield perfect (subjective) image quality at the lowest possible rate. Such rate is a fundamental limit that has been called *perceptual entropy* by Jayant et al in [46].

The ideal just noticeable distortion provides the image being coded with a threshold level of error visibility, below which reconstruction errors are imperceptible. Supposing that transparent coding cannot be achieved due to a tight bit-rate budget, rather than JND a supra-threshold generalization of such perceptual threshold would be required. The distortion presented in the reconstructed image is then called *minimally noticeable distortion* (MND) [46]. It should be minimally perceptible and (should appear to be) uniformly distributed over the image.

A coding algorithm based on the criterion of minimizing the perceived error is called a *perceptual coding* algorithm. The approach of taking into account the human visual system in the coding scheme is known as *second generation* image coding [54]. The essential task of perceptual coding is thus to effectively adapt the coding scheme to the sensitivity of the human eye. This allows the removal of perceptually redundant information for realizing high quality at low bit-rates. For obtaining moderate quality at even lower bit-rates, minimally perceptible image features will have to be removed by the coding system as well.

Measures of perceptual fidelity

To assess the quality of the reconstructed images, an effective fidelity criterion is needed. The peak signal to noise ratio (PSNR) is a widely used measure of image quality based on the computation of the mean squared error (MSE) between the original and the reconstructed images:

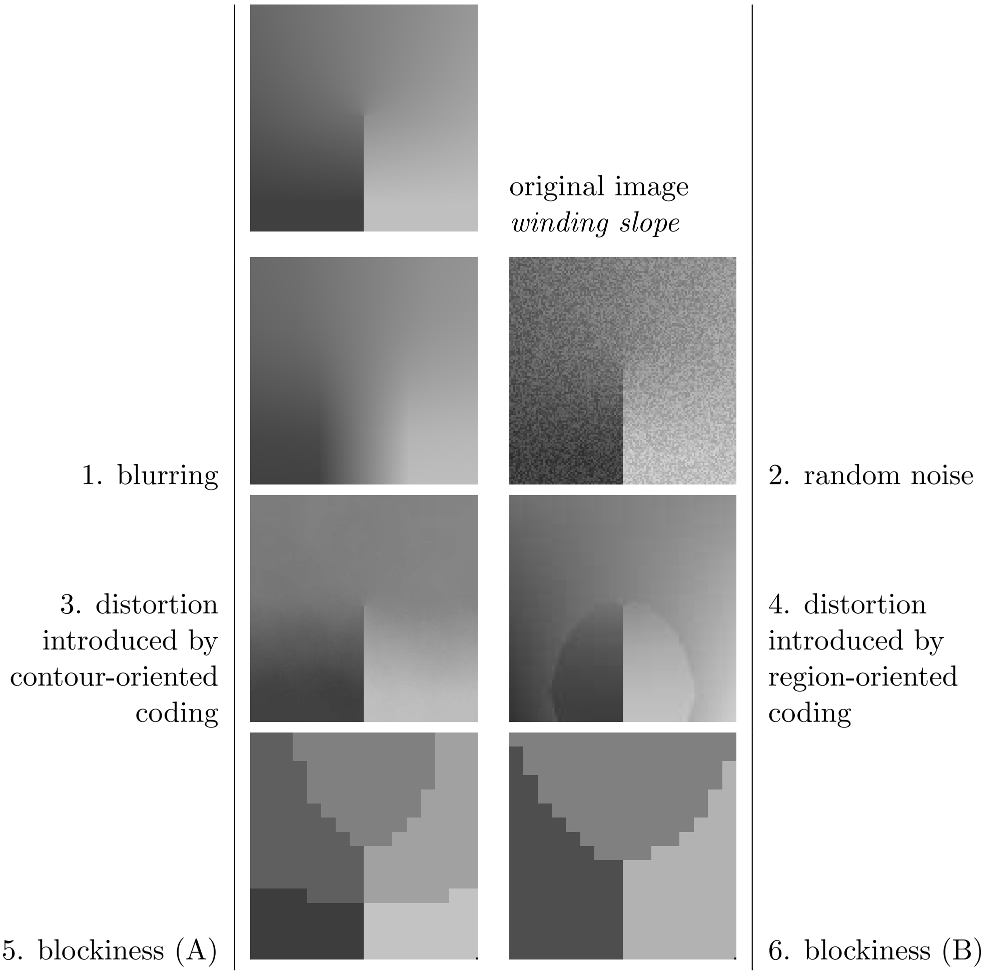
$$\mathrm{PSNR} = 20 \log_{10} \frac{255}{\sqrt{\mathrm{MSE}}} \tag{1.1}$$

It however cannot accurately reflect the *perceptual quality* of the reconstructed image –at least the simple computation of the PSNR for the whole image– and, particularly, at low bit-rates [62]. Nevertheless, the application of the PSNR measure in some contexts may be useful. Later in this chapter, the PSNR will actually be proposed as a valid measure of image fidelity in the context of perceptual image models. Some examples are in order here to illustrate the performance of the PSNR measure in its raw form.

The images presented in Figs. 1.1 and 1.2 are intended to show that the perceived distortion depends on the distribution and type of the introduced errors. In the first row of both figures two original images are shown: a synthetic image presenting a sharp radial transition, which will be called *winding slope* in the following, and a detailed area of the face from the popular image of *Lenna*. The six images displayed below, numbered from 1 to 6, present different types of distortion, namely: blurring, random noise, the distortion introduced by contour-oriented [12] and segmentation-based [94] coding techniques and two images showing block effects resulting from the JPEG coding algorithm [122].

Tables 1.1 and 1.2 show the PSNR values for the reconstructed versions of *winding slope* and *Lenna's face*, respectively. Whenever possible, the reconstructed images have been generated in order to get similar reconstruction PSNR values. This is the case for the first three images in both cases. When the distortion has been introduced by a particular coding scheme, the bit-rates measured in bits per pixel (bpp) are given.

Let us focus now on the subjective perception of distortion in the images shown in Figs. 1.1 and 1.2. From the observation of these images, it is worthwhile to realize the sensitivity of our visual system to the rendering of image contours. This is a property that has been



original image
winding slope

1. blurring

2. random noise

3. distortion introduced by contour-oriented coding

4. distortion introduced by region-oriented coding

5. blockiness (A)

6. blockiness (B)

Figure 1.1: Different types of distortion in a synthetic image

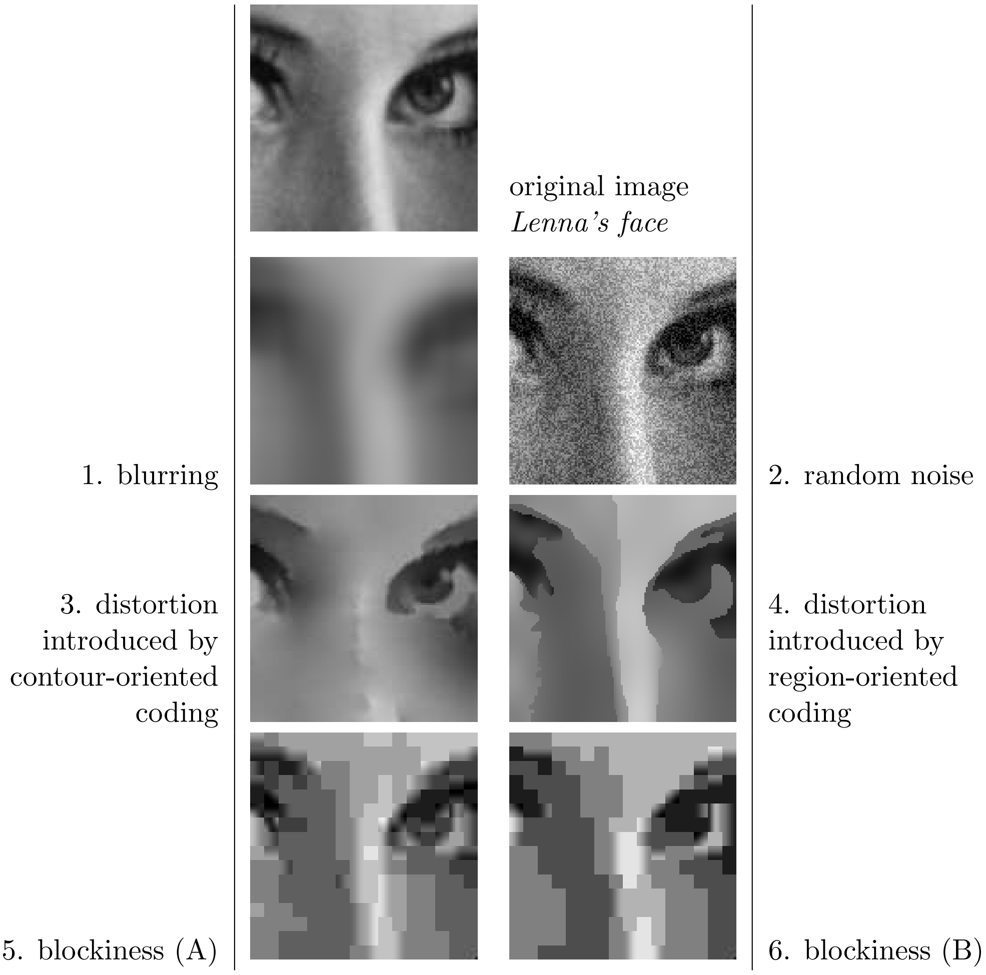

original image
Lenna's face
1. blurring
2. random noise
3. distortion introduced by contour-oriented coding
4. distortion introduced by region-oriented coding
5. blockiness (A)
6. blockiness (B)

Figure 1.2: Different types of distortion in a natural image

Table 1.1: PSNR values for Fig. 1.1

Type of distortion	PSNR value [dB]	bit-rate [bpp]
1. blurring	25.9	—
2. random noise	26.0	—
3. contour	26.0	0.03
4. region	35.9	0.04
5. blockiness (A)	28.4	0.30
6. blockiness (B)	24.8	0.29

Table 1.2: PSNR values for Fig. 1.2

Type of distortion	PSNR value [dB]	bit-rate [bpp]
1. blurring	20.6	—
2. random noise	20.8	—
3. contour	20.7	0.11
4. region	26.2	0.15
5. blockiness (A)	25.7	0.30
6. blockiness (B)	22.7	0.29

investigated in depth in psycho-visual studies [21], [66], in order to characterize the image features that are responsible for the formation of perception in the human visual system. In this sense, the blurred images (1) seem to give less information than the others. On the other hand, the presence of false contours (4, 5, 6) is a type of distortion rather 'confusing' for the eye. Furthermore, the visual system easily masks random noise (2) and large errors in smooth areas (3), distortions that do not cause excessive trouble.

For high distortion levels, such as those in the previous examples, it can be said that the PSNR (or the MSE) is not adequate as a measure of visual fidelity. In some cases, there is considerably discrepancy between subjective judgment and PSNR values (compare these values for images 3 and 4 or image 5 and images 2, 3). Specially noticeable is the PSNR value of the coded segmentation of Fig. 1.1. Nevertheless, the MSE is a commonly used fidelity criterion mainly for two reasons:

- its mathematical tractability and
- the fact that small values of MSE correspond to subjective high quality reconstructions.

Compression systems based on the minimization of the MSE measure by means of rate-distortion optimization algorithms have been reported with very good results for applications at high, moderate and low bit-rates using conventional image coding techniques such as vector quantization [105], subband coding [114], wavelets [84] and DCT [83].

At very low bit-rates, when significant quality losses are unavoidable, image coding techniques based on the minimum mean square error (MMSE) criterion usually produce specific types of visible distortion. The MSE criterion hardly discerns these coding errors from other and less visible distortion effects. The study of subjective measures of distortion seem to be the only answer in very low bit-rate applications.

1.2 Image compression based on human visual perception

The study of perceptual measures of distortion has been a main topic of research of the image coding community, hoping that a major breakthrough in image coding should rely on the use of such empirical measures. However, a second perceptual approach based on the definition of perceptual image models for visual signals seems to be more promising for very low bit-rate applications. This section is devoted to a brief overview of both. The two approaches have many points in common because both pursue an identical target. They are usually applied in a joint manner to image and video coding, and actually complement each other. Both points of view are explained separately in the following only for presentation purposes.

1.2.1 The point of view of perceptual measures of distortion

From the early years of image coding, a variety of methods have been proposed to incorporate certain psycho-visual properties of the human visual system into image compression algorithms [39], [73], [71], [19], [51], [20]. A comprehensive review of perceptual signal coding techniques has been presented by Jayant et al in a recent paper [46].

Some efforts have been directed to incorporate the sensitivity of the human visual system to spatial frequency components. Frequency sensitivity, is a *global* property dependent only on the image size and viewing conditions. It is described by the *modulation transfer function* (MTF) of the human eye, which has been obtained through psycho-visual experiments [21]. A similar property is contrast sensitivity. The human visual perception is sensitive to luminance contrast rather than to the absolute luminance [45], as indicated by Weber's law [71]. Both properties can be exploited via pre- and post-processing, by defining an *homomorphic model* [46] for perceptual coding such that, rather than weighting the distortion, the system weights the input and transforms it into a *(perceptually flat)* domain where an unweighted error measure is useful.

The main problem of the former approach is that it does not go far enough utilizing the masking properties of human vision. Our visual system is highly non-linear and presents important *local* properties such as *lateral inhibition* [21] which are difficult to measure. These properties depend on the local scene content, i. e., background intensity, activity of luminance changes, dominant spatial frequency and, in particular, the presence of important luminance transitions in the neighborhood [66], [117].

In order to achieve the highest subjective quality at a given bit-rate, all these properties of the human visual system must be exploited. The mapping of their effects to the JND/MND image profiles requires the existence of effective perceptual measures obtained from extensive subjective experimentation. The methods based on the JND/MND concepts have been very successful in transparent coding of audio signals [47], [49]. For image applications, proposals for visual fidelity criteria such as the *peak signal to perceptible noise ratio* (PSPNR) recently reported in [20], have been made. However, although such measures significantly improve the rendition of coding algorithms, no image coding scheme has yet sufficiently integrated these psycho-visual effects to offer a simple and efficient method to measure the coding impairment.

The target framework defined by Jayant et al [46] is based on thorough *perceptual distortion-rate functions*, so that the methods for rate-distortion optimization taken from the field of Information Theory could be applied straightforward in a perceptually flat image coding domain. Nevertheless, due to the lack of effective measures for the evaluation of image quality, this perceptual 'information-theory-based' approach is not likely to make significant advances in the near future on image compression algorithms.

Actually, most conventional coding methods use perceptual properties of the visual system; for instance, in the design of quantization tables for DCT coefficients. However, methods such as predictive coding, transform coding, subband coding or vector quantization are basically information-theory based, in the sense that image signals are considered as random signals and compressed by exploiting their stochastic properties. The high non-stationarity of image signals compromises the complete success of such conventional coding methods at low bit-rates, even if improved measures of perceptual fidelity are found. A qualitative change in the representation of visual signals is required in order to better exploit the perceptual properties.

Current standards for image compression [122], [112], [56] already exploit some aspects of visual perception but it is generally accepted that only the study of image models strongly related to the human visual system will lead to the highest compression values needed for very low bit-rate applications. These so-called perceptual image models and second generation coding techniques permit a graceful degradation of the perceived quality of reconstructed images at low bit-rates, without the unnatural artifacts (blockiness, ringing and blurring) of waveform coding techniques.

1.2.2 The approach based on perceptual image models

A promising approach to perceptual image coding is based on the definition of *perceptual image models*. Let us take the broadest sense of the word to mean by 'model' any perceptual 2-D or 3-D image coding model. Many different proposals have been presented so far that would fit within this large category, some of them relatively new in the image coding field.

Opposite to conventional coding methods which efficiently represent signal waveforms, perceptual image models represent image signals using structures which, in some sense, take into account the 2-D or 3-D physical (spatial) properties of the scene. A major advantage of these models is that they describe image contents by means of explicit structural features or elements of the model such as contours, regions and surfaces called image primitives. Some examples of 2-D perceptual models are those described in [54], [77], [12], [25], [1], [123], [97] and [85]. 3-D image models are often related to different application areas such as animation [33]. Some research groups are well-known from their work in the application of 3-D model-based techniques to image coding schemes. Overviews of 3-D model-based techniques techniques are given in [59] and [3].

A possible classification of coding techniques based on perceptual image models is reproduced in table 1.3. This table has been generated from the information in two original sources [59] and [2]. It only reports some sample examples and does not try to be a complete account

Table 1.3: A possible classification of coding schemes based on perceptual models

Source model	elements	image coding schemes	coding of sequence motion
region-oriented	regions	contour/texture coding	affine motion, warping
edge-oriented	edges	sketch/texture coding	primitive motion
mesh oriented	cells	mesh/texture coding	nodes motion
3D model-oriented	surfaces, volumes, parametric 3-D models	object-based coding	3-D global/local motion

of perceptual image models. Conventional waveform coding techniques could be included as well in this table, as these techniques make use of visual properties for image coding. However, it is difficult to justify that a single pixel –for instance in DPCM coding– or a square block of pixels –as used in vector quantization or transform techniques– can be considered as a visual primitive of a perceptual model of the image. Of course, many types of hybrid techniques are also possible and some are often used (i. e. block-based motion compensated prediction for inter-frame coding).

In perceptual image models, the items of visual information are mapped into elements of the model representing meaningful primitives in the original images such as edges, regions, objects or movements. The number and type of the extracted primitives is defined depending on the available bit-rate (quantization) and then fed to the entropy coder in order to extract the remaining redundant information. The extraction of the image primitives is often performed a priori –on a perceptual basis– according to empirical knowledge about image formation. Visual elements often considered significant are sharp transitions [12], homogeneous regions [97], contrasted objects over flat surfaces [1], high curvature lines [90], motion coherence [123], etc. The conventional source encoding process of mapping–quantization–entropy-coding [72] can be then improved by a coding scheme as given in Fig. 1.3, which refers to the application of the perceptual model in the representation (mapping) step of the visual information prior to the quantization and entropy coding steps.

Within the context of perceptual model-based representation, the design of perceptual measures of visual distortion is not such an important issue. For very-low bit-rate applications even if only a limited number of visual primitives or model elements can be synthesized

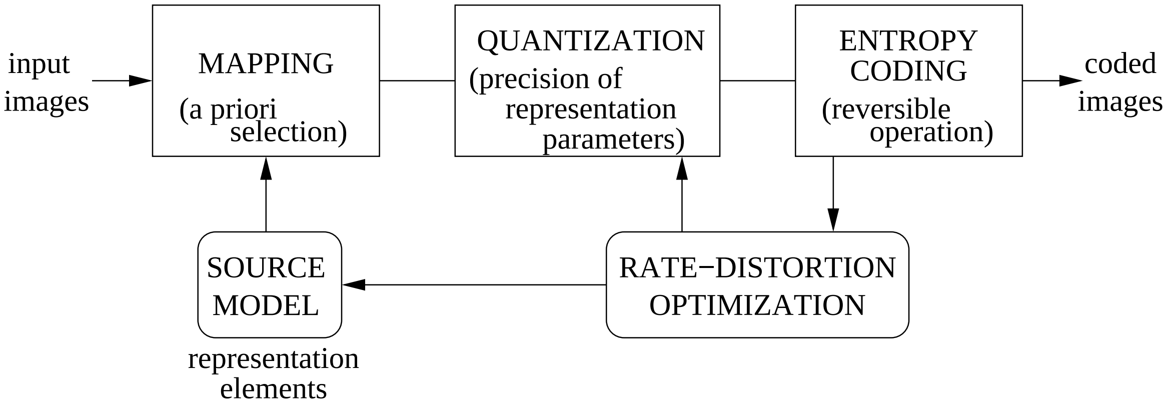


Figure 1.3: Encoding process based on a perceptual model of the image

in the reconstructed image, the resulting distortion artifacts appear to be more ‘natural’. At extremely low bit-rates [77] sketch-like, patch-work like or articulated-volume-like effects occur, often giving the impression of schematic views of the original images drawn by a painter or a cartoonist.

Coding schemes based on perceptual models even can make use of the ‘non-perceptual’ PSNR fidelity criterion for the rate-distortion optimization of the model-based representation [69]. The visually significant elements of the model used in the mapping stage have already forced the introduction of perceptual constraints, so that the distortion artifacts that may appear in the reconstructed image are better tolerated. For instance, when region-oriented image models are applied in segmentation-based coding schemes, the presence of contours in the reconstructed image is guaranteed by the partition structure, which is the main element of the image model. The PSNR measure in the (assumed to be) homogeneous regions is then perceptually significant.

1.3 Investigated approach

In the framework of this thesis, the characteristics of existing perceptual image models have suggested the study of related image and video coding techniques. The interaction of visual primitives, such as edges or regions of the perceptual models, with several conventional coding techniques has been investigated. The efficiency of waveform coding methods in exploiting stochastic properties of image signals, for instance, has been taken into account for the coding of homogeneous textures.

More precisely, we limit ourselves to the study and development of three *perceptually motivated* texture coding techniques that complement each other in a combined edge-oriented/region-oriented perceptual model of the image. The investigated texture coding techniques are, namely: *morphological interpolation*, which is based on image transitions taken as visual primitives from which smooth areas may be generated in edge-oriented models, *detail coding*, that looks at the efficient coding of small meaningful features for the human visual system, and *region-based subband coding*, aimed at the coding of fine homogeneous textures in the context of region-oriented models.

These techniques have been initially developed for the compression of grey-level and color still images and applied later for texture coding of both intra and inter-frames of video sequences within two different segmentation-based video coding schemes [97] and [22].

The reason for the study of such texture coding techniques is that most coding schemes based in perceptual image models have devoted more attention to the coding of their distinctive structural primitives –mainly regions and contours, in 2-D models– than to the homogeneous textures present in image data. Structural primitives are usually based on perceptually selected sub-signals, chosen because of their importance in the subjective process of image understanding. However, they may represent minority subclasses from the point of view of energy or from the point of view of the fraction of pixels involved. The performance of some conventional waveform coding techniques such as subband coding, achieving remarkable compression results, has not been fully exploited yet for the coding schemes using perceptual image models. This often happens because of the difficulties of adapting such conventional coding techniques to image models. The final balance of bit-rates, for instance, between contour (model element) and texture information in the coded images, shows the effective application of texture coding techniques in compression schemes based on perceptual image models.

1.3.1 Image model

The model of the image proposed for compression purposes is, in principle, an *edge-oriented* image model inspired in 'sketch-based' coding techniques [12], [25], [24], [86], [90]. The structural primitives of sketch models are *strong edges*, defined as alignments of 'high curvature pixels' of the image in the previously mentioned works. Sketch coding was first introduced by Carlsson [12][1] as an image compression method intended to represent grey level still images based on the coding of *geometric* and *grey level* information of the image contours (*sketch-*

[1] Carlsson cited a previous idea of Pearson and Robinson [76], [77] who proposed line drawings as economical but recognizable representations ('cartoons') for visual communications at very low bit-rates (4.8–19.2 kbit/s), with application, for instance, as visual aids for deaf people

data). Carlsson proposed an interpolation method allowing the reconstruction of a simplified (smoothed) version of the image from this information alone at high compression ratios (30–75). The resulting reconstruction could be improved by the coding of the residual texture by means of some waveform coding technique[2]. Sketch-oriented image models will be reviewed in more detail in chapter 2.

In the present work, the basic concept of sketch-based coding techniques has been applied in combination with a segmentation-based coding scheme, as a texture coding technique of the region-oriented image model. This means that both closed and open contours (regions and edges) are possible primitives of the image model. Closed contours define region boundaries, whereas open contours are the strong edges that will be taken as sketch-data. The characteristics of the particular image model will be discussed in more detail in chapter 2. Let us explain in the following the reasons that have led to the choice of this perceptual model.

Why strong edges

Images and video signals are highly non-stationary sources. They contain a wealth of segments of flat or slowly changing intensity, as well as edges and textured regions. In general [46], images may be characterized as being composed of large homogeneous regions –flat, sloping or textured areas– and strong edges having small spatial support. Pixel-to-pixel correlation is very high inside such regions, but not across the edges. The primary aim of image coding schemes is the extraction of this spatial redundancy. Pixel-oriented coding techniques working at low bit-rates generally present the problems near strong edges. They either fail to extract the redundant information or show 'perceptually' annoying artifacts such as ringing, blurring or jagged edges. This results either in low compression ratios or in low image quality for high compression applications.

The properties of the Human Visual System, in particular the special role of strong edges in our perception of images and their interaction with areas of smooth intensity variation, have been pointed out by researchers of the field of image perception [21] and early visual processing [66], [9]. An important function of coding algorithms from the point of view of visual perception is to render edge information faithfully. Perception-based image models and perceptual coding techniques have been proposed as a solution to this challenge. In particular, edge-oriented and region-oriented image models fully exploit the information about the edge structure of the images, its perceptual significance and the masking effect of sharp transitions. This produces clear subjective improvements over conventional waveform coding coding techniques, which are especially noticeable at low and very-low bit-rates.

[2] Carlsson proposed to use a Laplacian pyramid coding technique

Other components of the perceptual model

Strong edges and region boundaries are not sufficient for the reconstruction of high quality coded images. The coding result obtained from strong edges is called the *strong edge component* and it is valid only for high compression/moderate quality applications. However, the strong edge component images are rather smooth and lack textures. In the sketch-based coding schemes mentioned above, fine texture information is coded in a second (residual) component of the image model. Examples of texture coding techniques that have been employed are: pyramidal coding [12], wavelet coding [24] and DCT-based transform coding [86].

In this work, fine textures are coded as well from the residue of the strong edge component in the *texture component* of the proposed perceptual model. However, from the point of view of coding efficiency, a region-oriented image model may result in a better performance than a sketch-oriented image model. Segmentation-based coding schemes rely on a partition of the image in non-overlapping regions whose textures may be coded independently. Such coding schemes bring the possibility of developing improved waveform coding techniques for the coding of the residual texture information. In particular, we have adapted an efficient waveform coding technique such as subband coding for the independent coding of the individual contents of each region in a segmentation-based coding scheme.

Finally, it has been found that there is a particular type of image features which are not coded efficiently by any of the previous components. These features are small image details. They can be treated neither as textures –due to their small support and lack of homogeneity, periodicity, etc.– nor as contours –because of their short length and often isolated positions, what makes them expensive to code. A third component, the *detail component* has been defined to encode such features.

To summarize, a *three-component perceptual image model* is proposed. Each component of the model is coded by one of the investigated texture coding techniques:

- a *strong edge component*, coded by a morphological interpolation technique
- a *details component*, for which a purpose-designed coding technique has been defined,
- a *texture component*, coded by region-based subband analysis.

1.4 Organization and contributions

The organization of this thesis is as follows. Chapter 2 briefly overviews some perceptual edge-oriented and region-oriented image models to which the proposed coding technique is related. The 'structural' primitives defining such models justify the use of image analysis techniques strongly related to the physical image structure. Mathematical morphology provides an excellent set of processing tools that have been used extensively in the present work. In particular, appendix A is devoted to review the definition of some of the tools that will be used in the sequel.

Chapters 3, 4, 5 describe each one of the investigated texture coding techniques: morphological interpolation, detail coding and region-based subband coding. The application of these techniques is illustrated by means of various coding results on still images. Their advantages and drawbacks are discussed and compared to existing texture coding schemes. As subband coding is a well-known image coding technique, chapter 5 emphasizes its application in the framework of a region-oriented scheme for the coding of the contents of arbitrarily shaped regions. In appendix B, a set of filter banks proposed for subband analysis and synthesis are reviewed with special stress on the quadrature mirror filter bank. The reasons that have made us select these filters for the application to region-based subband coding are discussed in this appendix.

Chapter 6 presents the results of applying the above texture coding techniques to the coding of still images and video sequences in a segmentation-based video coding scheme that is briefly described in appendix C. Finally, chapter 7 is devoted to the conclusions and future lines of research.

Contributions of this work

To the best of our knowledge, the contributions of this work are summarized below.

- The proposal of a new interpolation method to perform spatial interpolation from arbitrary initial sets. Morphological interpolation is an interpolation technique based on a geodesic distance transformation intended to solve the problem of image interpolation from scattered data sets. It is more efficient in terms of computation time than linear interpolation techniques based on linear diffusion. Compared to other distance-based interpolation techniques, it has the advantage of diminishing its computation time as the number of initial pixels with different values increases, contrarily to interpolation techniques based on distance maps.
- The use of the watershed algorithm for the extraction of sketch data from the morphological Laplacian by means of the definition of a proper set of markers.
- The proposal of a explicit perceptual criterion for the selection of meaningful details in video sequences, including features such as shape, contrast, size, activity of the

background and temporal persistence.

- The adaptation of the Relative Address Element Designate (READ) code [130] for the coding of the position and shape information of image details in video sequences[3]. The proposed improvement consists in adaptively selecting the reference line among the previously coded lines in the current image or among the neighboring lines in the previous image (in temporal order).
- The proposal of a region-based subband analysis scheme for arbitrarily shaped image regions, along with the definition of region-based strategies for the bit-allocation, quantization and buffering of the analyzed subband data that fully exploit the edge structure of the segmented image.
- An effective modification of the symmetric signal extension technique proposed by Barnard [5] that results in a significant reduction of the variance of high frequency bands in the subband coding scheme. As a consequence, the high frequency subbands may be coded at lower rates and/or with smaller quantization errors.
- Finally, the proposal of a combined sketch-oriented/region-oriented model of the image for the coding of the textures of intra and inter-frames of video sequences in the framework of a segmentation-based video coding algorithm [22] that performs rate-distortion optimization in order to select an optimized partition structure and a set of texture coding techniques to be applied to the different regions.

[3] As explained in chapter 4, the READ coding is a differential run-length encoding technique proposed for the coding of binary images in facsimile transmission. It codes the runs indicating the positions of the black-to-white or white-to-black transitions in the current line differentially with respect to those of the previous (reference) line.

Chapter 2

Perceptual image models

Strong edges are sharp transitions of image signals that can be represented by geometric features (lines, curves). Such features are to a great extent responsible for the formation of perception in the human visual system [21]. The ability with which the eye interprets line drawings has been considered as a proof of the existence of special mechanisms in the visual system sensitive to *geometric structure* [9]. The images presented in Fig. 2.1 are intended to illustrate this point. These are 'sketch' representations extracted from original images that may be recognized only from their geometric structure.

The perception of geometric structures of images forms the basis of a theory of early visual processing put forward by Marr [66, ch. 2]. He proposed that, at the early stages of visual processing, the eye is able to extract visual primitives at different levels of resolution. Among these primitives Marr includes *edges*, *bars*, *blobs*, *termination points*, etc. that can be defined in very concrete or rather abstract ways (e.g. *zero-crossings* or *a cloud of dots*). They form what is called the *primal sketch*.

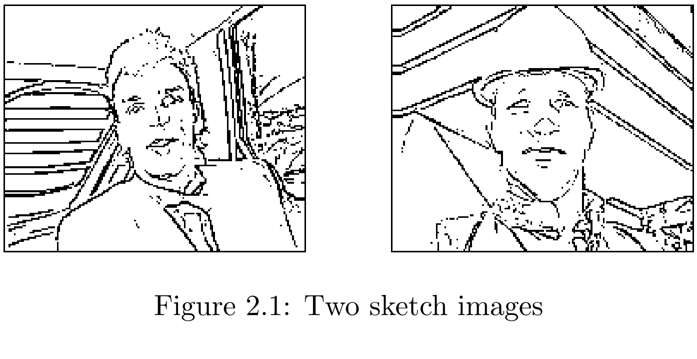


Figure 2.1: Two sketch images

The interesting suggestion made in the framework of Marr's theory is that the primal sketch is the only information that is used in subsequent visual processing. For coding purposes, this would imply that if a coding system is able to describe the primitives at the different resolution levels in the primal sketch, the image could be coded without perceived distortion. However, from the point of view of image coding, Carlsson [12] has questioned the completeness of a primal sketch representation. He argued that there exist non-trivial images that do not contain any contours as defined in the primal sketch.

Most model-based coding techniques rely on the description of visual primitives located at the position of grey level discontinuities (sharp transitions) in image signals. Region-oriented image models used in segmentation-based coding schemes, are aimed at the extraction of the objects' contours. Edge-oriented image models employed in sketch-based schemes code the transitions individually –without grouping them together to define regions. Mesh-oriented image models are used in coding schemes that tend to locate mesh nodes at points with high gradient values. The correct characterization of such transitions fulfills the important premise on which perceptual coding models are based: the matching of the image model to the visual perception process of the human visual system.

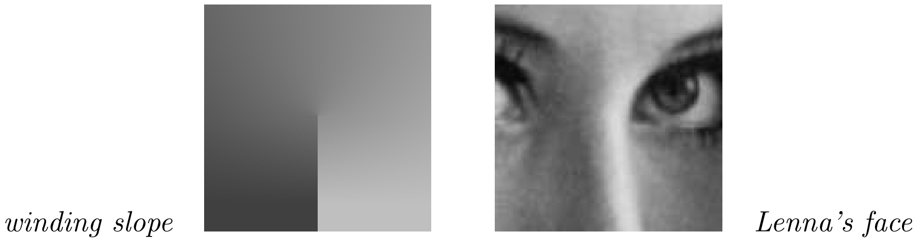
This chapter briefly overviews two types of image models based on the description of image transitions as structural primitives of visual signals. In particular, edge-oriented image models are compared with region-oriented ones, and several proposals of sketch-based coding schemes relying on edge-oriented models are reviewed. A three component perceptual model is put forward in section 2.2.

2.1 Overview of perceptual image coding models

The aim of this section is to compare two types of perceptual models for image coding: region-oriented and edge-oriented models. Elements of both models will be used in the model proposal described in section 2.2. Other perceptual image models not described here are, for instance, mesh-oriented image models [123].

2.1.1 Region-oriented image models

A well-known model of the image is the *region-oriented* model [54] resulting from segmentation techniques. A segmentation process *"divides an image into a set of homogeneous and connected regions related to the objects in the scene"* [64]. This results in a *partition* of the image usually represented by a label image, where each label corresponds to a different region. The perceptual approach of region-oriented coding techniques is based on the assumption that



winding slope

Lenna's face

Figure 2.2: Open contour features in synthetic (left) and natural images (right)

the regions describe objects perceived by the observer. In segmentation-based coding techniques the image is represented as a set of mutually exclusive spatial regions. The regions are separately coded as homogeneous distributions of textures. Their contents are either approximated by smooth functions (e.g. polynomial approximations) or coded using conventional waveform coding techniques [7]. The discontinuities in between are coded by means of contour following techniques [34], [65].

A dual representation such as the contour/texture description defined in a region-oriented model of the image presents some limitations. Objects can be found in an image that do not correspond to the basic idea of region employed in segmentation-based schemes. There can be very significant contours that are not necessarily closed. This can be seen, for instance, in the two original images of the examples given in chapter 1, which are repeated in Fig. 2.2. It is worthwhile to observe that similar features to the vertical open contour in the synthetic image *winding slope* are met in natural images as well, such as the shadows in *Lenna's face.* Of course, these images are very simple and the coded results that will be shown in the following cannot be extrapolated to the general case. They have been chosen, however, to better illustrate the advantages of edge-oriented image coding models in two examples where these models present clear advantages.

Open contour image features may not be represented accurately by regions of closed contours. The partition of the image resulting from a segmentation process requires all the contours to be *closed curves* in order to obtain mutually exclusive spatial regions. This may force the introduction of false contours. False contours often present random behavior due to the fact that these contours do not follow sharp transitions, what makes them strongly dependent of texture noise.

At very low bit-rates, false contours can make the coding process inefficient. This point is illustrated for the previous examples through the application of a segmentation-based coding

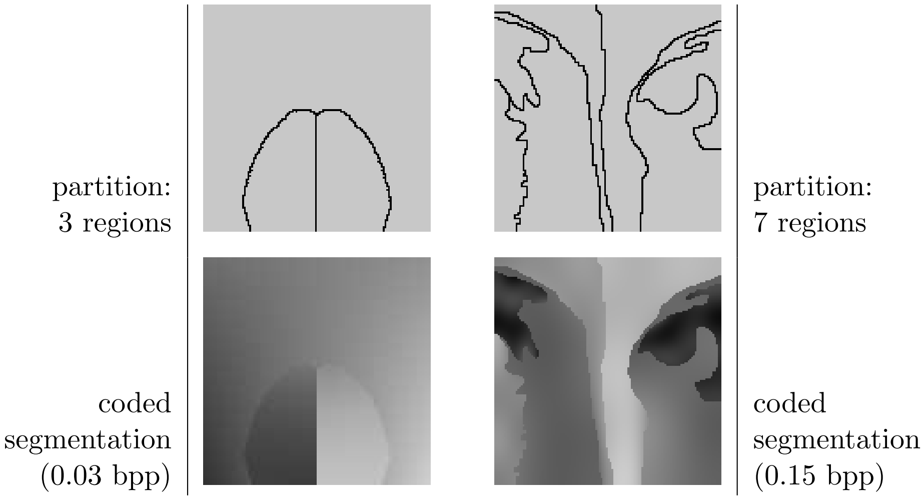


Figure 2.3: Partition results and low bit-rate coded segmentation

technique [94]. For both images, the partition resulting from the segmentation algorithm is shown in the first row of Fig. 2.3[1]. Notice that the segmentation has been forced to give only a small number of regions. If other segmentation algorithms with different homogeneity criteria had been applied, the resulting partitions could have been different from the ones presented in this figure. Nevertheless, if it is desirable to have contours located at the perceived edges of the presented images, as it is usually the case in order to apply texture coding techniques for the coding of the interiors of the 'homogeneous' regions[2] of the partition, some false contours would have resulted.

The coded segmentation is shown in the second row of Fig. 2.3. The interior of the regions has been coded with a smooth approximation, using fourth order cosine basis [37] (25 coefficients). The compression ratios are 200 (0.03 bpp) and 57 (0.15 bpp) for *winding slope* and *Lenna's face*, respectively. The constraint of mutually exclusive spatial regions forces the coding of some false contours. For instance, the two curves in the segmentation of *winding slope* or the contour crossing the cheek of *Lenna's face*.

For high compression applications, contours are expensive to code. To increase the coding efficiency, either the false contours should be simplified (or even removed) or more efficient

[1] The lines separating the partition labels have been drawn in order to better illustrate the location of contours. Actually, contour pixels are placed between every two pixels belonging to different regions, at a finer resolution.

[2] And, therefore, without sharp transitions in the interiors

(and complex) texture coding techniques for the coding of the inside should be used. This would also improve region-oriented coding schemes at lower compression ratios, where the usual problem is over-segmentation (excess of contours).

2.1.2 Edge-oriented image models

In order to have very efficient coding schemes, more flexible representations than a partition into regions should be introduced. The *edge-oriented* image models used in sketch-based coding schemes provide an alternative description for strong edges, which is less rigid than that of closed contours. The main differences of the coded edge information in sketch-based coding schemes are:

- Not only the spatial position of the transition is coded, but also the grey level or color gap it involves. This approach leads to a structural image primitive carrying both spatial and amplitude information about the transition.
- The constraint of closed contour is removed. Strong edges may be represented without the restriction (and the resulting cost) of being closed. However, this implies that contour tracing algorithms must introduce more initial and final points for the coding of such edges. A trade-off must be found between the cost of false contours and that of extreme points.
- Edge pixels are points located in the same grid of the original image. Therefore, contours are lines of pixels of the image.

The images *winding slope* and *Lenna's face* have been coded by means of a sketch-based coding scheme [12]. The results are shown in Fig. 2.4. The extracted sketch-data are the lines shown in the first row. In this case, both position and amplitude information are presented in these images. Amplitude values have been coded as first order polynomial approximations (lines with a certain slope) of the values of the original image along the extracted edge primitives. The reconstructed textures are obtained from the coded amplitudes at the transition points by means of an interpolation algorithm [13]. The compression values are similar to those of the coded segmentation, namely: 280 (0.02 bpp) for the first image and 71 (0.11 bpp) for the second one. Certainly, these examples are not representative of a general class of images. The coding results cannot be generalized and have to be treated with some care. Anyway, they illustrate the possibilities of the use of open contours as an interesting and promising representation based on edge primitives.

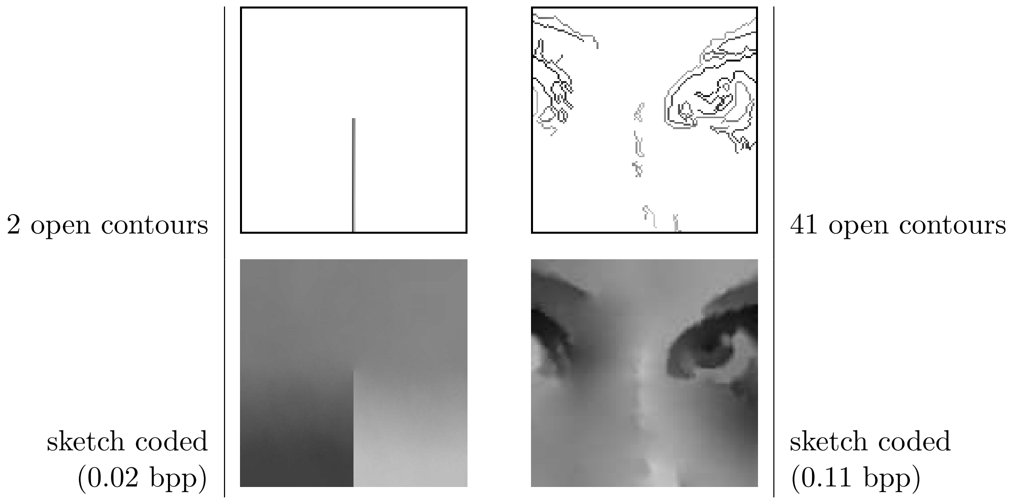


Figure 2.4: Sketch data and coded images

A number of sketch-based image coding schemes have been reported in the literature [129], [32], [12], [25], [24], [1], [86], [90]. The most representative in the context of this thesis are reviewed in the sequel.

An early sketch-based coding scheme for TV pictures

One of the earliest systems which may be considered as a sketch-based representation is due to Yan and Sakrison [129]. They reported a coding scheme for television pictures that efficiently encoded sharp transitions along horizontal scan lines. The amplitude values of each scan line were described by means of a two component model; the first component being a step-like discontinuous function and the second, the residual textures. The positions where the *breakpoints* of the discontinuous functions occurred were encoded in terms of run-lengths and the amplitude steps were quantized and entropy coded. The remaining textures were coded by means of a Fourier transform coder. Although this coding system did not thoroughly exploit the two-dimensional correlation of sketch data, the authors pointed out the usefulness of the separation in two components. They concluded that perceptually improved results could be obtained by coding image transitions separately in a explicit representation.

The two-component model of Carlsson

A well known sketch coding scheme is due to Carlsson [12]. He proposed a two component model for the coding of still images based on the sketch concept. The extraction of sketch information (sharp transitions) is performed by a Laplacian-Gaussian operator [66, ch. 2]. A contour following technique [34] codes the position information, with some geometric constraints in order to decrease the number of bits per contour point. Then, the grey level values for pixels on both sides of the edges are approximated with second order polynomials. The correlation of these pixels is exploited by using a differential coding scheme. A two-dimensional interpolation algorithm based on linear diffusion (Laplacian smoothing) performs image reconstruction from sketch data. Finally, the residual texture component is coded by means of the Laplacian pyramid coding scheme [10].

Carlsson showed coding examples with 'intelligible' reconstructed images for compression ratios in the range 65–75 (0.12–0.10 bpp). However, he found that in some cases irrelevant details and strong noise were strong enough to be extracted as contours, and the performance of the coding algorithm decreased. Carlsson suggested that only the perceptually relevant contours should be extracted to improve the performance of his method.

A three component sketch model

Some authors have considered the smooth texture information generated from the sketch data as a separate component of the perceptual model. This is the case for the model reported by Eddins and Smith [26]. They motivated their model by the problem of reducing edge ringing in subband coders. The components of this model are a severely down-sampled low-pass version of the original image and two high-pass components: the edge component, derived from the output of an edge operator along major image contours, and a texture component, formed by high frequency variations away from those contours. The discrete Laplacian is used as an edge strength operator. The edge amplitude profiles are coded by means of spline interpolation functions. An 'inverse' edge operator is defined to reconstruct the high pass frequencies of the image from the sum of the edge and texture information. Although the authors had not yet developed the edge component coder, they showed that the three component model was a flexible and complete image representation.

The 'perceptually motivated' model of Ran and Farvardin

Ran and Farvardin have investigated a three component image model as well [86]. The three components of the model, namely, primary, smooth and texture, are motivated by psycho-visual observations. Various experiments have been studied in order to mathematically formulate the interaction between strong edges and areas of smooth intensity variation [85]. A key concept of their work is the generation of a 'stressed' image by means of space-variant low-pass filtering. The results from such experiments suggest the characterization of strong edges as the high curvature energy pixels of the stressed image.

Contour tracing techniques have been proposed in order to locate the strong edges of the primary component by looking at high curvature pixels in the prefiltered (stressed) image. The authors claim that this strategy provides superior performance than the Laplacian-Gaussian operator for edge extraction. Besides, the space-variant low-pass filter is used as well in the generation of the low pass component from the coded edge information. This filter can be considered as an adaptive Laplacian smoothing operator minimizing a measure of 'energy variation' which corresponds to the curvature values at non-edge pixels.

For the coding of the primary component, edge positions are described by means of the Freeman code [34], whereas amplitude values along strong edges are quantized to a constant value. Ran and Farvardin have reported that this quantization gives rise to little perceptual degradation, assuming that the contour tracing algorithm breaks the contours at the points where a threshold variation is found. The smooth and texture components are added and coded together by means of an adaptive DCT coding scheme[3]. The authors suggest the separate coding of these components to improve coding performance, but they report that performance is similar to the coding of the sum because of the overhead needed for encoding the classification information for DCT quantizers.

Ran and Farvardin attribute the improved perceptual quality of the coding result of the three component model to the separate encoding of the primary component. It is worth noticing that the primary component in the coded images they show contains a small number of coded edges. Most of the contents of the image are thus coded by the adaptive DCT scheme as smooth/texture component. The adaptivity of a two stage classification of image blocks and the use of an efficient bit allocation algorithm contributed to yield a very good compression performance of their model.

An edge-based description of images

Contrarily to the above scheme, Grattoni [40] and Cumani [25] propose an edge-only representation of images. They propose an elaborate method of contour detection based on zero crossings of second derivative operators. Image reconstruction is done from the informa-

[3] Ran and Farvardin reported results employing a subband coding scheme as well, but the results achieved with the adaptive DCT coding were the best.

tion of positions and amplitudes of edge pixels by means of Laplacian smoothing. Although the reconstructed images lack of fine texture information these authors have proven that a (subjectively) faithful reconstruction is obtained from edge information alone.

A model based on 'ridge' and 'valley' primitives

Robinson [90] has defined ridge and valley primitives as the structural elements of the coding model. As in the previous case, this model consists of only one component and is intended for high compression purposes (around 50:1). The reconstructed images lack of textures but they present fair quality reconstructions from a perceptual point of view.

Robinson proposes the use of the extrema of the Laplacian-Gaussian operator as image primitives rather than Laplacian zero crossings. If the image amplitudes are seen as surface heights in a relief, the maxima of the Laplacian operators locate valleys, whereas the minima locate ridges. Robinson argues that, although ridges and valleys do not form closed curves, they are less noisy than zero crossings because the extrema of the Laplacian do not follow points of very rapid change in amplitude with respect to the position as zero crossings do.

Furthermore, this author employs a natural neighbor interpolation scheme for the reconstruction of the images from the sketch data [106]. He claims that this method produces smoothly interpolated images and it is ideal for this application, but very demanding from the computational point of view. Robinson concludes that more efficient interpolation schemes are needed for the problem of scattered data interpolation in the reconstruction of the smooth areas from sketch data.

2.2 Proposal of a perceptual image model

Discussion

The advantages of edge-oriented over region-oriented image coding models have been pointed out with the examples given at the beginning of section 2.1. Basically, edge-oriented image models provide a more flexible representation than the partition resulting from segmentation processes. This permits, on the one hand, to avoid the coding of false contours and to prevent false contour effects in the coded reconstructions. On the other hand, edge-oriented image models allow the generation of more complex smooth components by interpolation from sketch data than those resulting from smooth functions such as polynomials or low order cosine basis [37]. However, edge-oriented coding models present significant disadvantages with respect to region-oriented models. Let us mention some of them:

- An edge-oriented image representation is a lower level representation than the region-oriented one. This is stated in the sense that regions are intended to model objects, whereas edges 'only' describe image discontinuities. From the point of view of coding efficiency, there are cases where it could be better to keep the contours 'open', as has been shown in the previous examples. However, region-oriented schemes are most suitable to get a high level (object-based) representation of the image[4].
- A second drawback of edge-oriented models is found in the coding of motion in video sequences. Although some efforts have been made for the motion estimation (primitive-matching) and compensation of edge-like primitives [131] currently reported region-oriented motion estimation/compensation schemes seem to be more robust [99].
- A third drawback that can be argued against edge-oriented coding models is the fact that such representations do not permit the coding of homogeneous textures in separate regions. Instead, the textures are spread over the whole image in the residual texture component. This prevents, for instance, the local optimization of texture coding algorithms for individual regions containing distinct homogeneous textures. Therefore, an unequal distribution of bit-rates over different areas of the image is not so straightforward as in the region-oriented representation case.

These considerations lead to the following question: *is it possible to unify in a single framework the advantages of both region-oriented and edge-oriented models?*

Example

Up to the author's knowledge, the perceptually meaningful concept of a sketch representation has not been defined in the rigid scheme of a partition into regions. A possible application of the sketch concept in the framework of a region-oriented model is proposed in the following. It is illustrated for the same images *winding slope* and *Lenna's face* in the example of Fig. 2.5.

In this example, the partition structure is kept and the sketch data (edges) are defined as the pixels located at the boundaries of the regions. More precisely, the boundary pixels of each region form a boundary line which is broken into boundary segments at the points where a new neighboring region is encountered[5]. The mean value of the region is computed

[4] In line with content-based descriptions currently investigated for the forthcoming MPEG4 coding standard [70]

[5] That is, at pixels located at the region boundaries having neighboring pixels that belong to more than two different regions.

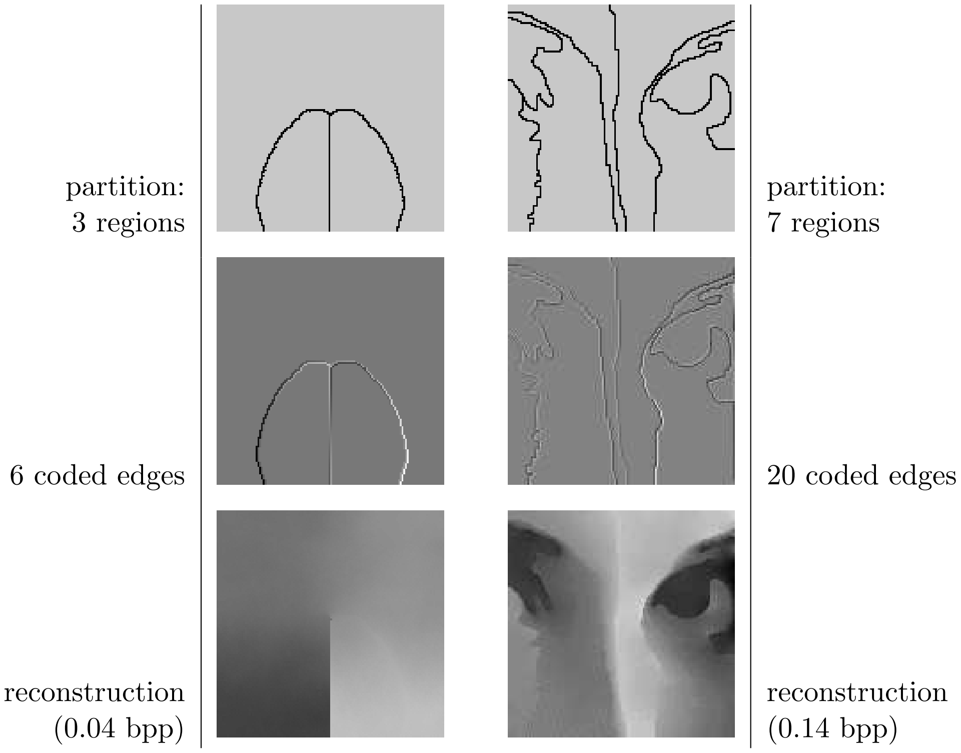


Figure 2.5: Example of application of the sketch concept within a partition

and the difference between the pixels located at each boundary segment and this value is approximated by a second order cosine function. The images in the central row show such coded differences[6]. The interpolation from the approximated amplitudes at the boundary segments yields the results shown in the bottom row of Fig. 2.5.

Please notice that the effect of false contours is greatly diminished in this reconstructed image compared to the coded segmentation results that were shown in Fig. 2.3. The improvement may be explained from the fact that only information at the region boundaries is used. Therefore, edge amplitudes are correctly adjusted whereas the reconstruction may yield large errors in smooth areas. As reported from psycho-visual observations [21], such errors are less noticeable if the contour information is correctly represented. On the contrary, the coding

[6] Middle grey corresponds to zero level; these images have been stretched from -64 to +64 in order to better illustrate the variations along the boundary segments.

of textures by means of smooth polynomial or cosine functions equally approximates all the points of the region.

For the example of Fig. 2.5, the cost of the coded sketch representation has been kept at the same rate than the coded segmentation of Fig. 2.3. The cost of the coded amplitude values is actually small in smooth regions (recall that only the differences with respect to the mean of the region are coded) and a little larger in complex regions. This example shows that interpolative coding from the amplitudes of the pixels located at region boundary segments may be applied as a texture coding technique in a segmentation-based coding scheme. The compression is not as efficient as in the sketch-based representation presented in Fig. 2.4 but, in compensation, the potential advantages of a region-oriented representation are kept. An additional advantage is that rather complex low-frequency textures can be generated by interpolation from the sketch information only.

Model proposal

The aim of this section is to describe the choice of a particular proposal for a perceptual image model, taking into account both the experiences in our own research and the advantages and disadvantages of the above referenced perceptual image coding models. The perceptual model that is proposed consists of three components. Namely a *strong edge component*, a *details component* and a *texture component*. The coding techniques studied in chapters 3, 4 and 5 are related to this model proposal. Such techniques will be applied as texture coding techniques in the framework of a segmentation-based video coding scheme.

- **Strong edge component**
  This component of the model is based on the sketch concept of interpolative coding. The sketch data (edge information) is extracted from the original images at the location of sharp transitions. From this initial information alone a reconstruction of the image is obtained by means of an interpolation technique. Such reconstruction is able to efficiently represent complex textures in some areas of the image and, in addition, it decreases the visibility of false contours. A computationally efficient *morphological interpolation* method will be proposed in chapter 3.

  The strong edge component is applied to the coding of still images and intra-frame images in video sequences. Examples of sketch coding using both open and closed contours will be shown for still images. In the case of video sequences, we will restrict the application of the strong edge component in a segmentation-based coding scheme with regions of closed contours. This choice is motivated by the availability of region-based motion estimation and motion compensation coding schemes, whereas the motion compensation and coding of edge-like primitives has not yet reached the maturity of the analysis of motion in region-oriented models. In inter-frame images, it has been found that, when the prediction error shows edge-like features, the interpolation from region boundary segments also gives good coding results.

- **Details component**
  Small image features are sometimes lost in low bit-rate coding systems because they are supposed to be the least significant information for the observer. However, some of these features may be of great importance for the subjective judgment of the coded images. In video-telephone sequences, for instance, some facial shapes and shadows are especially significant even being smaller or dimmer than other features of the image. However, if all small features were to be coded, their cost would be rather high. It would imply the accurate transmission of high frequency components in the texture component or a large number of small contours in the strong edge component. Nevertheless, a number of meaningful small features must be coded –even at very low bit-rates– in order to match their perceptual significance. Failure to code these features would certainly affect the perceived quality of the reconstructed images.

  The details component has been devised for the extraction and selection of small features in image and video signals. The selection is based on an explicit criterion relying on the perceived parameters of such features. It takes into account parameters such as the shape, contrast, size, background activity and temporal persistence of the extracted details. An efficient coding algorithm for the coding of the details' positions is employed for this component. In inter-frame mode, significant details are tracked through consecutive frames along the time and their positions coded with reference to the previous ones. The detail coding technique is presented in chapter 4.

- **Texture component**
  The third component of the model is computed as the residue of the coding of the two previous components. At this stage, only fine homogeneous textures remain. A waveform coding technique (subband coding) has been adapted for the coding of such textures in a segmentation-based coding scheme. The new technique, explained in chapter 5 is known as region-based subband coding and may be applied both for inter and intra-frame images.

  At this point, it can be argued that region-based subband coding could be applied as well for the coding of smooth textures of the strong edge component, making unnecessary such component. However, this choice has the drawback of presenting larger ringing effects[7] at the boundaries of the region than the proposed one.

[7] A typical distortion of subband coders

Chapter 3

Image coding using morphological interpolation

Interpolative coding techniques are based on the coding and transmission of a subset of pixels of the original image so that, on the receiver side, the remaining pixels have to be interpolated from the transmitted information alone [72]. The reconstructed image is usually approximated by smooth, continuous functions with some permissible error at the interpolated positions. The subset of transmitted pixels, called the *initial set* in the following, may be either a regular sampling grid or any arbitrary set of points. In the latter case, both the amplitudes and positions of the pixels of the initial set should be coded and transmitted.

The application of interpolative techniques to image coding relies on the selection of a proper set of initial pixels. The initial pixels should, at the same time, allow a good restoration of the image by interpolation and lead to a compact representation. Given that interpolation methods only generate smooth surfaces, the initial pixels must be selected so that the main transitions between these surfaces are kept. Obviously, regular sampling grids are not suitable for this purpose. The selection of the initial set in an interpolative coding framework is a problem of data-dependent sampling of two-dimensional signals. For a given reconstruction (interpolation) method, this problem can be stated as follows:

- What are the best locations at which to take samples of a given image? Here *best* means that the image interpolated from those samples is as close as possible to the original.

- What is the minimum number of samples required to represent the image such that it can be reconstructed by interpolation within a given error tolerance?

The problem of finding the optimal initial set is an ill-posed problem because the solution is data-dependent, matched to the particular input image. Global numerical optimization is impractical, given the large number of candidate structures. Robinson and Ren have investigated the above questions in a recent paper [91] concluding that this problem should be addressed by means of an heuristic shape-driven search, tending to place initial samples in the neighborhood of significant image transitions. More precisely, they suggest that second derivative extrema to some extent characterize the features of interest. Strong edges are image features that give rise to extrema of the second derivative[1]. According to Robinson and Ren, the locations of such features should be taken as starting points for optimization algorithms aimed at the solution to the problem of selecting the samples of the initial set.

Link to perceptual image models

The above considerations link interpolative coding techniques and most perceptual models that have been proposed for image compression. Generally speaking, the problem of image coding derives precisely from the fact that images, in general, contain sharp transitions. As discussed in chapter 2, sharp transitions represent non-stationarities in the image where traditional waveform coders, designed for the extraction of spatial redundancy in homogeneous (stationary) areas, do not work properly. Furthermore, sharp transitions –strong edges– play a key role in our perception of visual information. For such reason, perceptually motivated image models are especially aimed at the explicit representation of these non-stationarities.

In *segmentation-based* coding schemes, the explicit representation consists of the *geometric* description of the contours of the 'objects' present in the image. Contours resulting from a segmentation process are closed curves, usually placed along the main image transitions, that define a partition of the image into a set of regions. The contents of the regions are coded by means of texture coding techniques.

Based on the importance of strong edges in the visual perception process, *sketch-based* coding schemes employ a model of the image relying on strong edges. The idea behind the sketch description is that, at very low bit-rates, textures between contours do not have to be represented explicitly in order to obtain an intelligible reconstruction of the original image [77]. The image is assumed to be mainly made of areas of constant or smoothly changing intensity separated by the discontinuities produced by strong edges. Under this assumption, a grey level image can be reconstructed from the sole information about the geometric structure of the transitions and the *amplitudes* of the transition pixels.

[1] Actually, Robinson and Ren describe luminance 'surface' features such as *edges*, *valleys*, *ridges* and *roofs* as good candidates for sample placement algorithms. In a later paper [90], Robinson has reported a coding scheme based on ridge and valley image primitives that proves the adequacy of these features for image representation.

Therefore, in sketch-based coding schemes the coded information consists of the shapes of the discontinuities and the values of the pixels located on both sides of such structures. In practice, this information, also known as *sketch data*, may be efficiently represented by coding, for instance, the position, width and height (amplitude step) of the significant image transitions. The reconstruction is posed, then, as a problem of *scattered data interpolation* from arbitrary initial sets –the sketch data– under certain smoothness constraints.

The aim of this chapter is to present a fast interpolation algorithm, called *morphological interpolation*, intended to perform spatial interpolation from any set of initial pixels. After reviewing various existing methods proposed for scattered data interpolation, the new technique is described in section 3.3. Morphological interpolation is based on morphological (non-linear) operators, namely geodesic dilation and the morphological Laplacian. This operators may be efficiently implemented in order to obtain fast reconstruction from sparse initial sets. Morphological interpolation is more efficient in terms of computation time than linear interpolation techniques based on diffusion processes, which apply iterative space-variant filtering to the initial image. Diffusion processes have been widely used for interpolative coding from sketch data. Comparative figures of computation time will be given to assess the efficiency of the morphological interpolation technique.

Several strategies for the selection of the initial set are investigated in section 3.4 at the end of the chapter. The application of morphological interpolation is illustrated by means of a sketch-based coding scheme aimed at very low bit-rates. Finally, a cost-effective image representation based on networks of lines is proposed. Networks of lines are interesting because they can be efficiently coded using derivative chain code techniques. The coded images will be proposed as well as the strong edge component of the perceptual model for image compression that has been introduced in the previous chapter.

3.1 The interpolation problem

The problem of spatial interpolation from arbitrary initial sets can be stated as follows.

Let I be a two-dimensional grey scale image containing the initial set S, as shown in Fig. 3.1. $D_I \subset \mathcal{Z}^2$ denotes the definition domain of I:

$$I \quad \left(\begin{array}{rcl} D_I \subset \mathcal{Z}^2 & \rightarrow & \{0, 1, \ldots, N\} \\ p & \mapsto & I\,(p) \end{array} \right.$$

The points of the initial set, $S \subset D_I$, are supposed to take discrete values in the range $[0, N-1]$, N being an arbitrary positive integer, and all other pixels p are arbitrarily set to

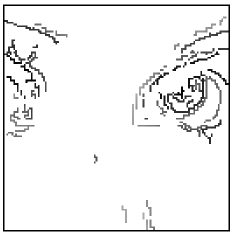


Figure 3.1: An example of initial set

the highest value N:

$$I(p) = \begin{cases} s_p & \text{if} \quad p \in S \\ 0 & \text{otherwise} \end{cases}$$

The purpose is to determine the numerical values at the interpolated positions (the pixels set to N) by using the known values at the points of the initial set. In other words, we want to find an exact interpolant $R : D_I \rightarrow \{0, 1, \ldots, N-1\}$, such that:

$$R(p) = \begin{cases} s_p & \text{if} \quad p \in S \\ r_p & \text{otherwise} \end{cases}$$

The interpolant must satisfy a number of conditions which, among other things, make it relate naturally to the initial data and make it reasonably smooth, so that there is a good chance that the reconstructed image looks very much like the original one from which the initial values where drawn. Sibson [106] has outlined some desirable properties of any scattered data interpolation method:

1. The interpolant should be at list continuously differentiable (C^1). C^1 functions are visually smooth and have smooth contour lines. Functions which are not continuously differentiable do not look smooth. Higher-order smoothness properties than C^1 do not appear to be detectable by the eye except in special cases.

2. The dependence of the interpolant on the initial values should be very simple; it is better if it can be actually linear, so that if, for example, all the initial values are multiplied by a scalar, the interpolant is also multiplied by that scalar.

3. The dependence of the interpolant on the positions of the initial pixels should be reasonably well-behaved; at least, continuity is desirable, so that the interpolant does not jump from one state to another in response to a small change in the values or positions of the pixels of the initial set.

4. The interpolant should be localized, in that in some suitable sense only initial pixels which are reasonably near neighbors should influence the interpolated value at a given point.

5. The method should be computationally feasible on a reasonably large scale. Localization can allow very large problems to be split and the results fitted together.

6. Finally, one should expect the interpolation method to recover exactly some simple functions such as constants, first degree functions and perhaps quadratic functions. The more ambitious interpolants in this sense are less localized.

3.2 Existing interpolation methods

Interpolation from sketch data is difficult using simple linear interpolation filters because of the high dependency of the result with respect to the spatial distribution of the initial pixels. For instance, cubic B-splines in two dimensions [82] are attractive candidates for image interpolation because of their properties of continuity and smoothness at sample points. Splines do perform well for regular sampling grids, but they should be adapted by some means to the geometry of an arbitrary distribution of sample points. An overview of some techniques proposed to solve the problem of scattered data interpolation is given below.

Linear diffusion methods

Several authors [12], [40], [1], [85] have proposed methods based on successive over-relaxation as a solution to the interpolation problem. These methods consist of an iterative smoothing by means of linear filtering. The values of the pixels of the of the initial set are kept unchanged through the filtering process. The evolution of the image with the successive iterations along the time can be described as a discrete approximation to the heat conduction (or *diffusion*) equation or by means of partial differential equations (PDE's) [100]. The progressive smoothing is interpreted as a diffusion process whereby the amplitude values at the initial points are diffused into the areas to be interpolated. This fact has an interesting connection with curve/surface evolution theory in computer vision. It has been shown that the number of extrema of a given function can never increase over the time provided that the evolution is governed by the heat equation [52]. This means that new structures cannot be generated and the interpolated image will only contain the contours present in the initial set, thus satisfying the smoothness constraint imposed to the interpolation [12].

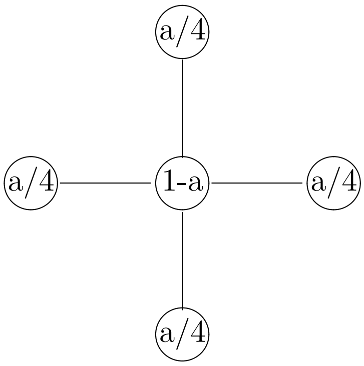


Figure 3.2: Typical kernel used for diffusion processes

Linear diffusion methods result in very smooth image reconstructions from the initial set. However, the main drawback of these methods is the high computational load, which strongly depends on the configuration of the initial set. The practical implementations make use of iterative space-variant filtering operations that converge rather slowly to the final interpolated image. A typical filter kernel used for interpolation by diffusion methods is shown in Fig. 3.2. Convergence is guaranteed for values of the relaxation parameters $0 < a < 2$. Computation times of more than one thousand seconds on computer workstations have been reported for the interpolation of a single 256 × 256 image [85]. In order to speed up the convergence of the iterative smoothing algorithm, Carlsson [12] proposes the use of multi-resolution grids to reduce the average distance between the points of the initial set in the first iterations of the diffusion process.

Methods based on distance transformations

A second approach to the interpolation problem is to use a purely geometric process, as the one proposed in [110]. This is a contour-specific technique based on distance transformations. It was initially developed to interpolate topographic surfaces from level lines in order to produce raster spatial distributions of terrain altitudes, called digital elevation models (DEM's). In this framework, the mentioned technique results particularly efficient. Efficient algorithms were proposed in [110] for implementation. The author reported execution times of 200 s for the interpolation of a entire 512 × 512 input image of contour lines. In addition to the improvements in computational efficiency, this technique produces better spatial distributions than other geometric processes intended for the interpolation of DEM's.

The interpolation based on distance transformations is computed by a linear combination of the values h_i of the connected components of constant value of the initial set. These values are weighted by the inverse of the distance d_i from the connected components to the interpolated position. That is [110], [95]:

$$x = \frac{\sum_i h_i/d_i}{\sum_i 1/d_i} \tag{3.1}$$

This equation assumes a constant slope of the interpolated function along the line linking a pixel to the components of the initial set. The distance function was proposed to be computed by means of geodesic distance transformations in a Euclidean metric, what enabled the precise calculation of distance values for all the pixels to be interpolated whatever the structure of contour lines.

In practice, the implementation of this 'geometric' interpolation algorithms relies on the generation of a *distance map* starting from each connected component. However, if there is a large number of such components in the initial set (or if the pixels of the connected components do not have constant amplitude values), the computation time spent in the generation of the distance maps from each connected component of constant value may be rather large.

The method of inverse distance weighting was proposed as an interpolative coding technique for image compression in [68]. A generalization used to estimate a grey level function starting from an arbitrary set of connected components of constant value (and not only from the level lines of the original function) has been reported in [95].

Finite element methods

Another solution to the interpolation problem relies on a finite element approach [106]. The original image can be split up into polyhedral cells containing the initial points, for instance using Dirichlet tessellation/Delaunay triangulation algorithms. Then, the interpolation can be performed by fitting together smooth functions across the cells, known as finite elements or *shape functions*.

The technique employed for the reconstruction of the cells of the 'active mesh' scheme reported in [123] in intra-frame mode is an example of interpolation based on the finite element approach. In this example, the nodal positions and their pixel values form the initial set. This method is suitable for initial sets consisting of isolated points, where a tessellation is straightforward, but not for arbitrary initial sets with groups of connected pixels of different amplitudes, as in the image shown in Fig. 3.1.

Natural neighbor interpolation

Other methods intended to solve the problem of scattered data interpolation from arbitrary initial sets have been discussed by Sibson in [106]. Methods such as kriging [8] or natural neighbor interpolation also present the problem of a high computational load. Using natural neighbor interpolation, Sibson reported computation times larger than 10 seconds for raster sizes of only 25 × 25 [106, p. 33].

In particular, the *natural neighbor interpolation* method has been used by Robinson for image coding in [90], and he concludes his paper proposing the development of more efficient interpolation schemes as a future line of research.

3.3 The morphological interpolation technique

The target of the morphological interpolation algorithm is to approximate the amplitudes of the unknown pixels of the image by fitting a surface on a subset of pixels of known values (the initial set). Such surface is constrained to be maximally smooth between the known pixels, in the sense that pixel to pixel variations in the interpolated area should be minimized.

A suitable strategy for spatial interpolation from sparse sets is the geometric approach of the methods based on distance transformations. Assume that the initial set is composed of the connected components R_i of known amplitude values (h_i) shown in the schematic representation of Fig. 3.3. As defined by eq. 3.1, the interpolation at the unknown points x may be computed as the average of the amplitudes of the connected components weighted by the inverse of the distances d_i to each of them. With such weighting, the amplitudes of the nearest components have stronger influence than those of the distant ones, and the interpolated amplitudes change slowly in the areas in between.

3.3.1 Geodesic distance weighting

The use of *geodesic distance* transformations has been proposed for the distance weighting factors d_i of eq. 3.1 [110]. The geodesic distance is defined in this case within the set of unknown pixels, i. e., as the length of the shortest path joining two points which is completely included in this set. An important advantage of the geodesic distance for interpolative coding purposes is that it allows the preservation of the transitions imposed by the initial set. This is illustrated as well in Fig. 3.3. Let us suppose that the brightest component R_2 represents the upper edge of a spatial transition. The darkest component R_3 represents the lower edge. The

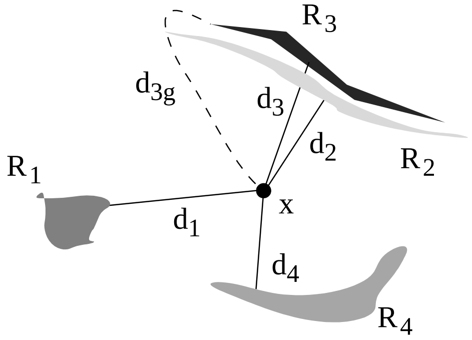


Figure 3.3: Interpolation of pixel x by inverse distance weighting

influence of the amplitude values of the lower edge at pixel x is given by the inverse of the geodesic distance d_{3g} (dashed line), which is larger than the Euclidean distance d_3. Therefore, the interpolated value at pixel x will be mainly influenced by the initial pixels of the upper edge (component R_2) given that the weights assigned to R_3 –located on the other side of the transition at a larger geodesic distance– will be much smaller. As a result, the use of the geodesic distance allows the preservation of the transition indicated by the two components R_2 and R_3, corresponding to the upper and lower edges.

3.3.2 Two-step iterative algorithm

As pointed out in section 3.2, interpolation methods based on distance transformations require the generation of a number of distance maps equivalent to the number of connected components of constant amplitude value present in the initial set. For an initial image as the one shown in Fig. 3.1 (p. 36), where amplitude values are not constant between neighboring pixels, the implementation based on distance maps is rather inefficient. Furthermore, the larger the number of pixels contained in the initial set, the longer the computation time required for the interpolation, what does not seem to be reasonable.

An alternative interpolation algorithm has been investigated as a solution to the problem of spatial interpolation in such cases. *Morphological interpolation* approximates the result of inverse distance weighting methods by means of an efficient two-step procedure. The proposed implementation handles at the same time both position and amplitude information of the pixels of the initial set. It performs an intuitive smoothing of the areas to be interpolated without the need to generate distance maps (which were obtained from position information only). The resulting interpolation is not so smooth as in the linear diffusion case but, as will be shown in the examples, the smoothing is sufficient for application in the framework of

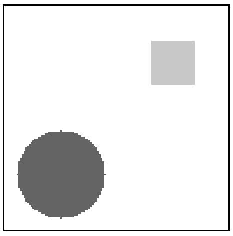


Figure 3.4: A simple initial set with two components of constant value

interpolative coding techniques.

Starting from the set of initial pixels, the two steps of the morphological interpolation algorithm, namely *geodesic propagation* and *smoothing*, are successively iterated until convergence. A very simple example of initial set has been chosen in order to illustrate the description of the algorithm. It is shown in Fig. 3.4, and consists of two small geometric figures of constant grey level. Of course, this is an easy case for the interpolation based on distance maps. It has been chosen for a more clear illustration of the morphological interpolation algorithm and to make possible the comparison of the performance of both methods. The morphological interpolation technique will be applied afterwards to the initial set that was presented in Fig. 3.1. Later in this chapter, examples with more complex images will be shown.

- *Geodesic propagation step*
  Instead of computing maps of geodesic distances from all the unknown pixels to every point of the initial set, the amplitude values of the known pixels are propagated 'geodesically' to fill the empty areas of the image. Fig. 3.5 shows several intermediate images corresponding to the geodesic propagation step.

  The geodesic propagation is implemented by means of a FIFO queue. First, the initial image is scanned in order to find all the 'empty' neighbors of the initial pixels (at geodesic distance 1). The grey level value of the neighbor in the initial set is given to each one of these pixels and then its position is put into the queue. If the pixel happens to have more than one neighboring initial pixel, the amplitude is chosen randomly among them. During the propagation, one pixel is extracted from the queue and its amplitude is propagated to all the empty neighbors whose locations in turn are put into the queue. The process stops when the queue is empty. Therefore, each pixel is treated only once in order to perform a complete geodesic propagation.

- *Progressive smoothing step*
  At the positions where two or more propagation fronts originated from initial pixels

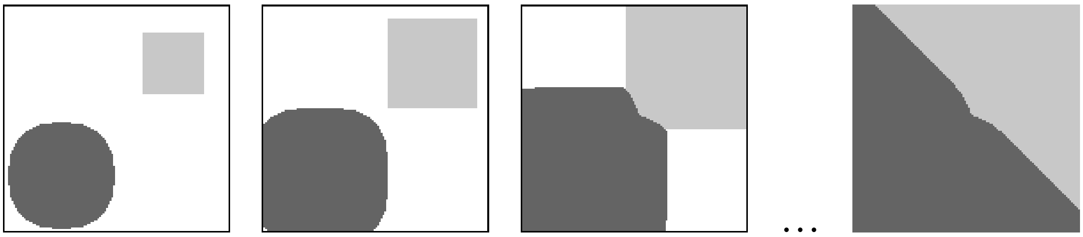
of different amplitudes meet, the process stops and a false transition is created. The false transitions appearing outside the set of initial pixels are smoothed in the second step. The morphological Laplacian[2] is used as a transition detector in order to obtain these false transitions. Pixels on both sides of the false transitions compose the set of *secondary pixels.* A grey level value equal to the average of the intensity values on both sides of the transition is then assigned to each secondary pixel. This is the smoothing step. Secondary pixels will be used in the next iteration of the algorithm in order to smooth out these transitions.

The position of the false transitions is actually known a priory from the positions of the initial set[3]. In a more complex initial set where these components may not have constant amplitude values, the morphological Laplacian performs a more efficient detection of the false transitions than if they were to be tracked through the propagation process. In addition, a threshold may be employed to locate only the false transitions with a large amplitude step.

- *Iteration*
  Then, a second iteration is performed: the propagation step propagates the grey level values from the sets of initial *and* secondary pixels. The propagation creates new false transitions which define a new set of secondary pixels where grey level values are smoothed again. Note that this new set of secondary pixels generally does not include the previous secondary pixels. This process of 1) propagation of values from initial and secondary pixels, and 2) smoothing of the grey levels at the false transitions, is iterated

[2] See appendix A for the definition of the morphological Laplacian

[3] In such a simple case as the initial set of Fig. 3.4, the false transitions form the geodesic 'SKIZ' of the complement of the initial set, i. e. the boundaries of the geodesic zones of influence of the connected components of the initial set.



distance 4 distance 12 distance 24 ... (distance 72)

Figure 3.5: Intermediate images and result of the geodesic propagation

until convergence. Fig. 3.6 shows several iterations of algorithm. Please observe the progressive smoothing of the false transitions. After a few number of iterations, the algorithm quickly converges to the final interpolated image.

3.3.3 Evaluation of the interpolation results

The interpolation result presented in the last row of Fig. 3.6 is very similar to the results obtained with other methods. For comparison, the interpolations with linear diffusion and inverse distance weighting (through the generation of distance maps) are shown in Fig. 3.7. To better appreciate the similarity, the difference images are shown in Fig. 3.8, stretched from ±32 to ±128 and shifted by 128. The differences are partly due to numerical errors (the calculations have been made with integer values) and to the fact that the morphological interpolation algorithm does not iterate until idempotence. It is stopped when the difference between two iterations is smaller than a given threshold. The variances of the difference images are given on top of each image. Taking as a reference the linear result, the morphological interpolation and the interpolation obtained with distance maps are the most similar.

The interpolation for this simple example is obtained more efficiently with the methods based on distance transformations than with the proposed algorithm and, clearly, in both cases more efficiently than in the linear diffusion case. However, this simple initial set image was only chosen to illustrate the description of the algorithm. The initial sets of interest would rather be like the one presented in Fig. 3.1, i. e. composed of linear features whose pixels do not have constant amplitude values. In such case, the methods based on the generation of distance maps would not be very efficient. The techniques employed in interpolative image coding with these images (sketch data) are usually based on linear diffusion algorithms.

The two steps of the morphological interpolation algorithm for the initial image shown in Fig. 3.1 are illustrated in Figs. 3.9 and 3.10. Notice that the propagation fronts generate surfaces with a certain degree of shading, due to amplitude changes along the lines of the initial set. In this case, interpolation algorithms based on distance transformations should have computed as many distance maps as connected components of constant value exist along each line. On the other hand, the morphological interpolation algorithm can be applied as before, and convergence is attained even with a smaller number of iterations than in the previous example of Fig. 3.4.

The interpolation result is compared in Fig. 3.11 with the interpolation resulting from linear diffusion techniques. Some slight differences can be observed between these images. It can be said that linear diffusion interpolation looks ‘smoother’ than morphological interpolation, but the smoother look of linear diffusion results has been reported by some authors

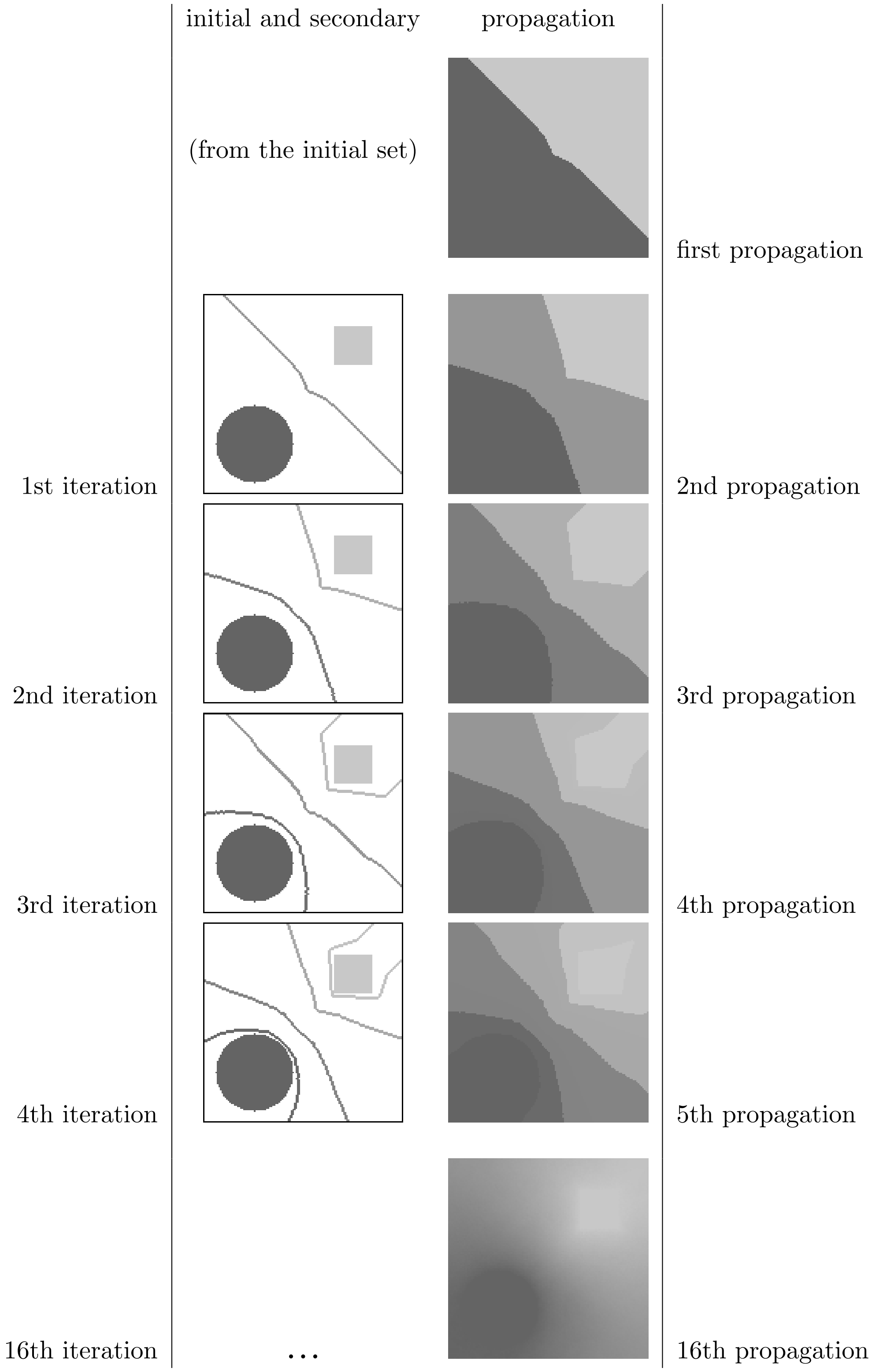


Figure 3.6: Smoothing iterations: left, initial and secondary pixels; right, propagation

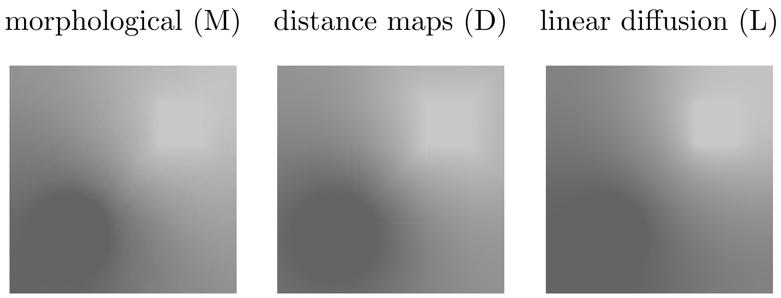


Figure 3.7: Comparison of interpolation techniques

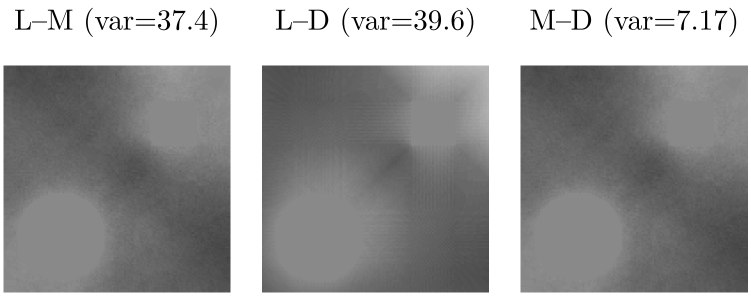


Figure 3.8: Differences between the interpolated results of Fig. 3.7 (the images are stretched from ±32 to ±128 and shifted by 128)

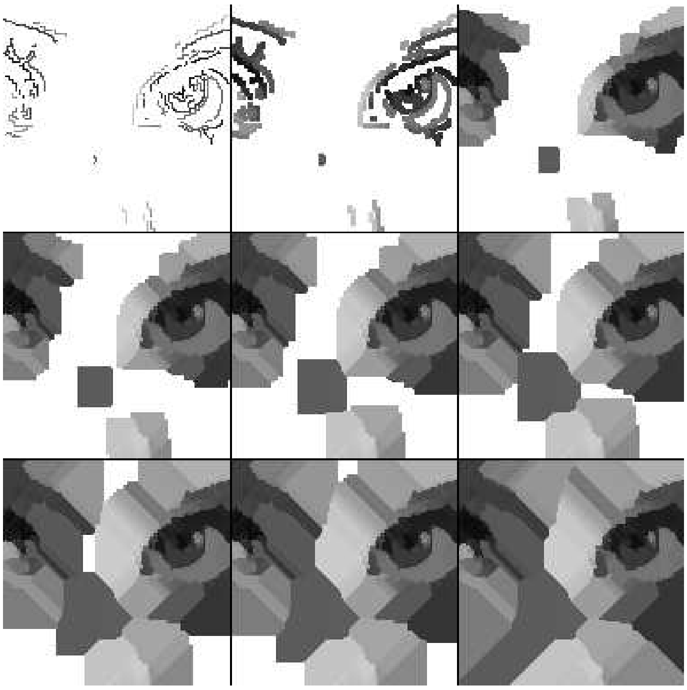


Figure 3.9: Geodesic propagation in a natural image: initial pixels and intermediate images

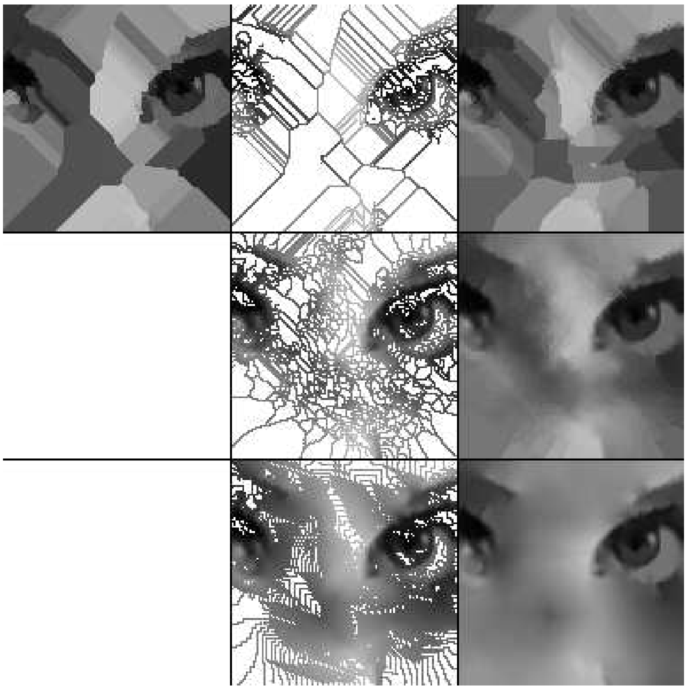


Figure 3.10: Three iterations (2nd, 4th and 8th) of the progressive smoothing step in a natural image. Left image: result of the first geodesic propagation, middle column: initial and secondary pixels, right column: geodesic propagation from these pixels

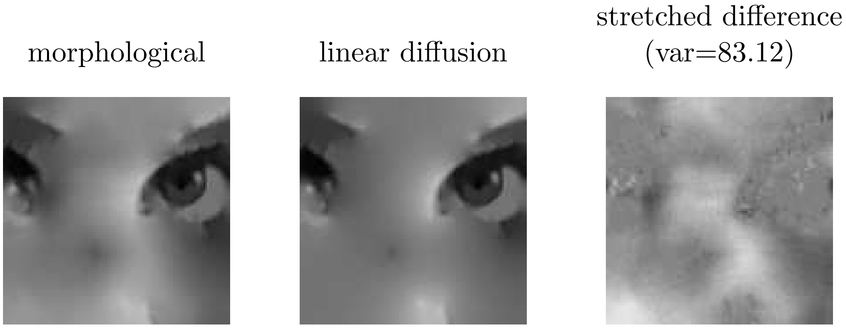


Figure 3.11: Comparison of morphological interpolation and linear diffusion results

[24] as a reconstruction artifact called ‘china doll’ appearance. The morphological interpolation algorithm has been forced to stop the iterations before idempotence in order to avoid such undesired effect. Apart from this, the results obtained with both techniques are very similar from the perceptual point of view in most images. Therefore, for interpolative coding purposes, the slight differences in reconstruction quality are not a distinctive property.

Algorithm efficiency

The ability to handle simultaneously the positions and amplitudes of the pixels of the initial set permits the application of the morphological interpolation method on any arbitrary distribution of pixels. Actually, the computation time decreases as the number of initial points increases. This behavior is also found in interpolation algorithms applying linear diffusion. However, morphological interpolation is much faster than linear diffusion algorithms.

The efficiency of the morphological interpolation algorithm in terms of computational load is illustrated in tables 3.1 and 3.2. Comparative figures of execution time[4] are given for the previous examples, both for the morphological interpolation algorithm and for interpolation by linear diffusion. Please notice the drastic reduction in the number of iterations needed for the morphological technique. Each pixel of the image to be interpolated is treated hundreds of times less. Furthermore, each iteration of the morphological interpolation does not require any multiplication, decreasing the time of each individual iteration compared to the linear filtering technique. This explains the reduced execution time of the described non-linear interpolation process. Clearly, there is no need of multi-grid techniques for speeding up convergence when

[4] Note: CPU times were computed on a Sun SPARC10 workstation

Table 3.1: Execution times of morphological interpolation and linear diffusion (Fig. 3.6)

Interpolation technique:	Execution time [sec]	No. of iterations
linear diffusion	312.8	4980
multi-grid diffusion	53.3	equivalent to 795
morphological interpolation	2.8	16

Table 3.2: Execution times of morphological interpolation and linear diffusion (Fig. 3.10)

Interpolation technique:	Execution time [sec]	No. of iterations
linear diffusion	458.3	2980
multi-grid diffusion	60.3	equivalent to 376
morphological interpolation	2.4	13

the morphological interpolation algorithm is used.

3.4 Interpolation and sketch image coding

Several approaches using interpolative coding techniques for ‘perceptually motivated’ compression applications have been reviewed in chapter 2. Perhaps the most well known being those of Carlsson [12], Ran and Farvardin [86] and Robinson [90]. The underlying image model is based on the perceptual concept of the ‘raw primal sketch’ [66]. The coded information (sketch data) consists of the geometric structure of the sharp transitions and the amplitudes at the edge pixels of such transitions.

For very low bit-rate applications, the decoder has to reconstruct the smooth areas of the image using only the coded sketch data. The reconstruction process is performed from this arbitrary initial set by means of a scattered data interpolation technique. As interpolation algorithms are designed to approximate the areas spanning among transitions by ‘smooth’ functions, they do not render the fine textures adequately. For coding applications at higher bit-rates, the residual texture information is separately coded by means of a waveform coding technique, for instance, pyramidal, transform or subband coding, in a different component of the image model.

The performance of such perceptual model has been thoroughly investigated [86], proving its utility for most coding applications and showing subjective improvements over DCT-based methods, such as JPEG, at low bit-rates. However, one of the important drawbacks is the large computation time spent in the interpolation process. Morphological interpolation is proposed as an efficient alternative for fast reconstruction from sketch data that gives similar interpolation results.

3.4.1 The problem of finding the optimal initial set

Robinson and Ren [91] have developed optimization algorithms aimed at the solution of the distortion-constrained and sample-constrained problems of interpolative image coding. The *distortion-constrained* problem consists in finding the minimum number of sample points necessary to obtain a reconstruction of the original image by interpolation within a given error bound. The *sample-constrained* problem can be stated in a similar way: to seek out the positions of a constrained number of samples that yield the optimal interpolated reconstruction in the mean squared error sense.

The solution to these problems obviously exists, but it is hard to obtain. Global optimization of the initial set is infeasible, due to the huge number of possible combinations that should be tested, even for a small image. The solutions proposed by Robinson and Ren are heuristic-driven search schemes. In the sample-constrained case, starting from a given sampling structure, they propose a sequential algorithm that moves sample positions in order to improve monotonically the signal to noise ratio (SNR) of the interpolated image. In the distortion-constrained case, an iterative sample removal procedure is used to minimize the number of samples for a given SNR. These algorithms provide good results with confidence of near optimality.

The observation of the behavior of sample placement optimization algorithms confirms that transition points are important for the sampling of a two-dimensional signal. The best candidate sampling structures (initial sets) for image interpolation are located at points with largest curvatures. Actually, the extrema of the second derivative is often a superset of the initial pixels resulting from these optimization algorithms [91].

However, a scattered sample representation relying on these results cannot be used straightforward for an economical representation of the image. In terms of raw data, the storage required for the amplitude values and locations of the pixels of the initial set would be larger than the one required for just the values of the original image.

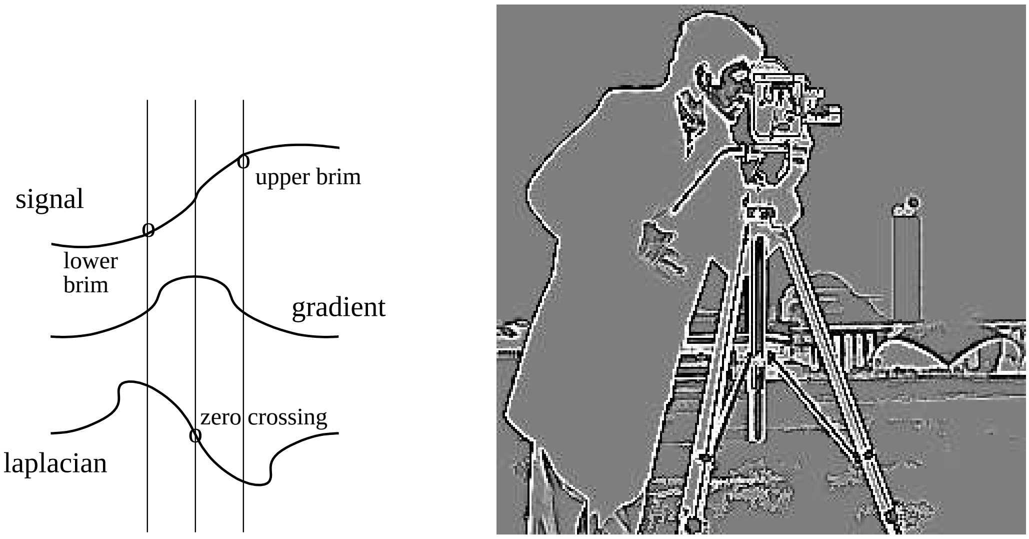


Figure 3.12: Location of edge brims using Laplacian extrema. Left: upper and lower brims. Right: morphological Laplacian of the cameraman image (Note: mid grey corresponds to zero level)

A sample experiment

As pointed out in appendix A, an estimate of the signal second derivative can be computed using the morphological Laplacian. This is a non-linear approximation to the Laplace operator in continuous space that was first studied in [116] for edge detection. The morphological Laplacian is greater than zero at the lower edge of the transitions and smaller than zero at the upper edge. It cancels out in flat surfaces or slanted planes without convexity changes.

The extrema of the second derivative locate the points with largest curvature values. These points occur at the upper and lower sides of the transitions, bringing information about the transition width and the intensity change. The left drawing in Fig. 3.12 is an illustration of the one-dimensional case. In the two-dimensional case of the image *cameraman* (Fig. 3.12, right), the set of points where the morphological Laplacian reaches significant values mainly corresponds to the perceived image contours.

The strategy of selecting pixels with large curvature values for interpolative coding was proposed by Carlsson in [12], and has been employed later by several researchers [25], [15], [13], [90]. The results of the previously mentioned optimization algorithms also suggest the validity of such strategy for the initial set.

The following experiment has been carried out for a close examination of this idea. In the left image of Fig. 3.13, a set of pixels having absolute values of the morphological Laplacian above a certain threshold is shown. If we attempt to reconstruct the rest of pixels of the smooth areas in between, the result will be the one presented in the right image. Morphological interpolation has been used as the reconstruction technique but in this section we will concentrate on the selection of the pixels for the initial set.

About one tenth of the pixels of the original cameraman have been used as initial pixels for the interpolation result shown in the example. The peak signal to noise ratio (PSNR) of the interpolated image is only 23 dB but the subjective quality is not bad. This may be explained because our attention is primarily drawn to the strong transitions which have been correctly placed and reproduced.

This experiment suggests that it is possible to obtain a smooth approximation of the original image from the amplitudes and positions of pixels having large curvature values. Furthermore, the morphological Laplacian performs as an effective enhancement operator for the detection of such set of initial pixels. Some alternatives of initial sets will be proposed in the following which are more suitable for coding purposes than a simple thresholding of the Laplacian image.

3.4.2 Application to the coding of the primary component

The former example has shown the possibility of performing interpolative coding from a set of initial pixels with large Laplacian values. However, the application of this idea to image coding relies on the selection of a proper set of initial pixels. The *initial set* should lead to a compact representation and, at the same time, allow a good approximation of the original image by interpolation. In the current section, two coding strategies using morphological interpolation are presented. In the first one, the initial set consists of isolated points, whereas in the second one the components of the initial set are networks of lines.

Coding by isolated points

This example deals with an image representation involving isolated points. Here, the objective is to select the smallest number of points leading to a good restoration of the image. One possible solution consists in using an iterative selection process.

In the first iteration, the pixels of absolute maximum and minimum amplitudes are se-

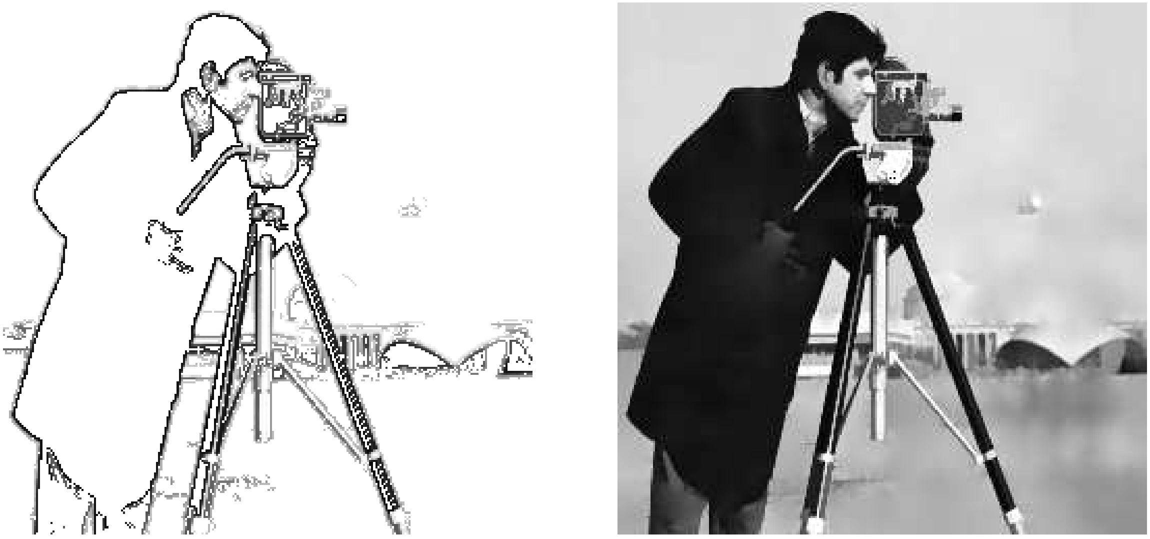


Figure 3.13: Morphological interpolation from pixels with large curvatures: left, initial image (10% pixels); right, interpolation result

Table 3.3: Compression ratio for the example of coding by isolated points

Iteration	No. of points	Compression
1	10	451
10	55	125
25	93	75
50	183	40
75	270	30
100	382	20

lected. A first reconstruction by morphological interpolation is performed and the residue with the original frame is computed. From this residue, a second set of maximum and minimum points are selected. They are used together with the first set of pixels to compute a second restoration. This process is iterated in order to reach a sufficient quality of the restored image. Fig. 3.14 illustrates various iteration steps. The iterative selection process has been performed using the small image of *Lenna's face* to avoid the computational load of the repeated interpolations from a small number of initial points in a larger image. In this case, the computational load increases exponentially with the image size, and even morphological interpolation would require large interpolation times. The whole simulation of 100 iterations for this small image was performed in a CPU time of 165 s in a SPARC10 workstation. The iteration numbers of the presented images are respectively of 1, 10, 25, 50 75 and 100. For each iteration, initial pixels, interpolated image and residue are shown. Notice how isolated points are introduced and the progressive quality improvement in the interpolated image.

The coding of the positions of the isolated points is performed by an Elias code [29]. The amplitude values are simply stored in a buffer following the scanning order and entropy coded by arithmetic coding [126]. Table 3.3 gives the number of isolated points together with the compression ratios for the reconstructions shown in Fig. 3.14.

Coding by maximum and minimum curvature lines

The drawbacks of the former strategy are, on the one hand, the high computational load of the iterated reconstruction processes –even using morphological interpolation– and, on the other hand, the fact that for more complex images requiring a larger number of initial points for a faithful reconstruction, the representation consisting of isolated points may not be efficient at all. Nevertheless, this example has shown that the set of initial points leading to a good

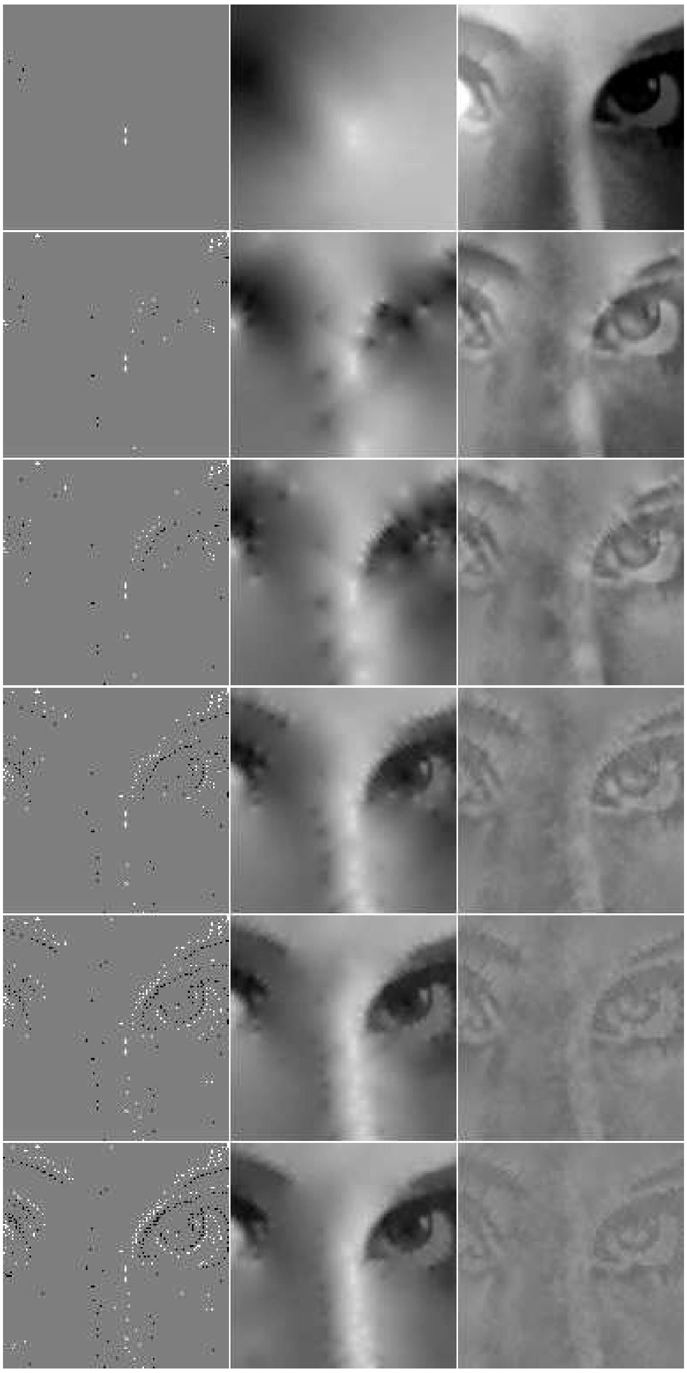


Figure 3.14: Morphological interpolation from a set of isolated points. For each row: initial set (left), interpolated image (center) and residue (right). Compression ratios: 451, 125, 75, 40, 30, 20

restoration of the image by interpolation are somehow aligned in the areas of largest curvature values, as suggested by the experiment of Fig. 3.13. The advantage of the placement of initial samples in these areas was also inferred from the behavior of the optimization algorithms studied by Robinson and Ren [91].

A natural coding strategy could be to group these points in a network of lines. Networks of lines are interesting for coding because they can be efficiently coded using derivative chain code techniques. The lines of largest curvature are called upper and lower *edge brims* by some authors [85]. Edge brims may be be obtained as the 'crest' and 'valley' lines of a second derivative operator. These lines do look promising for the characterization of visual information from a perceptual point of view. Robinson [90] claims that edge brims are less noisy than Laplacian zero-crossings, which follow the transition midpoints. Edge brims do not show so many random fluctuations because they do not represent a very rapid change in value with respect to position as transition midpoints do.

Extraction of 'edge brims' using the watershed

The edge brims of an image may be detected by computing the *watershed of the Laplacian* and of its dual with an appropriate set of markers. The watershed operator is one of the major decision tools in mathematical morphology. It is aimed precisely at the detection of the crest (or divide lines) of the image, seen as the surface of an imaginary relief. A large number of algorithms have been proposed for the efficient computation of the watershed. The most efficient ones are based on immersion simulations and rely on hierarchical queues. The reader is referred to [121] for further details on the watershed algorithm.

In order to obtain the edge brims, the watershed is applied twice to the Laplacian image with a set of markers formed by the union of two of the following sets:

- markers of the connected components of negative Laplacian values
- markers of the connected components of positive Laplacian values
- markers of the flat areas of the original image larger than a given size

In the case of the cameraman image, the morphological Laplacian has been already shown in Fig. 3.12 (right). The sets of markers are presented in Fig. 3.15. For the extraction of the lower brims (divide lines of the Laplacian), two first sets of markers are used (flat areas and Laplacian valleys). For the upper brims (valley lines of the Laplacian), the markers are formed by the union of the first and the third set (flat areas and Laplacian peaks) and the watershed is applied on the dual of the Laplacian image. The last image of Fig. 3.15 shows the result of the two applications of the watershed. The white and black lines correspond, respectively, to the crest and valley lines of the Laplacian or, likewise, to the positions of the lower and upper edge brims of the initial image. Please notice that some pieces of contour have been removed from the watershed result either because the Laplacian was not significant enough at these positions or because the lines were too short. The necessary thresholds have been chosen on an empirical basis. This result can be seen as a simplified version of the Laplacian image of Fig. 3.12 with the advantage that it can be coded efficiently.

If the initial set is composed of the pixels at the positions indicated by the watershed lines shown in Fig. 3.15 and the intensity values are approximated with first order polynomials, the interpolation results in the right image of Fig. 3.16. The reconstruction PSNR is in this case 20 dB, only 3 dB smaller than that of Fig. 3.13.

The geometric structure of the brim lines may be coded at low cost by means of a contour-following technique [65]. The amplitudes of the initial pixels in these lines must be coded also with a few number of bits. Given that intensity values along edge brims should keep rather constant, a simple approximation may be employed to code the values within each brim line. In the current example, a derivative chain code technique [65] has been used to code the pixels' positions. The starting points of the open contour chains are separately coded by means of an Elias code. The amplitude values have been coded by means of a polynomial approximation. More precisely, the network of brim lines is broken at each triple point (points with more than two branches). Then, the amplitudes of the pixels located under the resulting curves are approximated by a first order polynomial. The two coefficients defining each polynomial are quantized, entropy coded and transmitted.

In the example of Fig. 3.16, the overall bit-rate is 0.17 bits per pixel (bpp). The number of brim 'pieces' is 132: 72 upper brims and 60 lower brims (displayed in black and white in the last image of Fig. 3.15). Table 3.4 gives the proportion of this rate spent in the coding of position (shape and starting points of coded brims) and amplitude information[5].

A second example of coding by brim lines is the interpolated image *Lenna's face* used to illustrate the morphological interpolation algorithm in section 3.3. The extracted brims were shown in Fig. 3.1 and the interpolation result in Fig. 3.10. In that case, the reconstruction bit-rate was 0.11 bpp (21% amplitudes, 59% shape and 20% starting points).

[5] The number of bits per pixel (bpp) shown in the table has been computed by dividing the number of bits employed by the *total* number of pixels of the image, even in the case of contour information

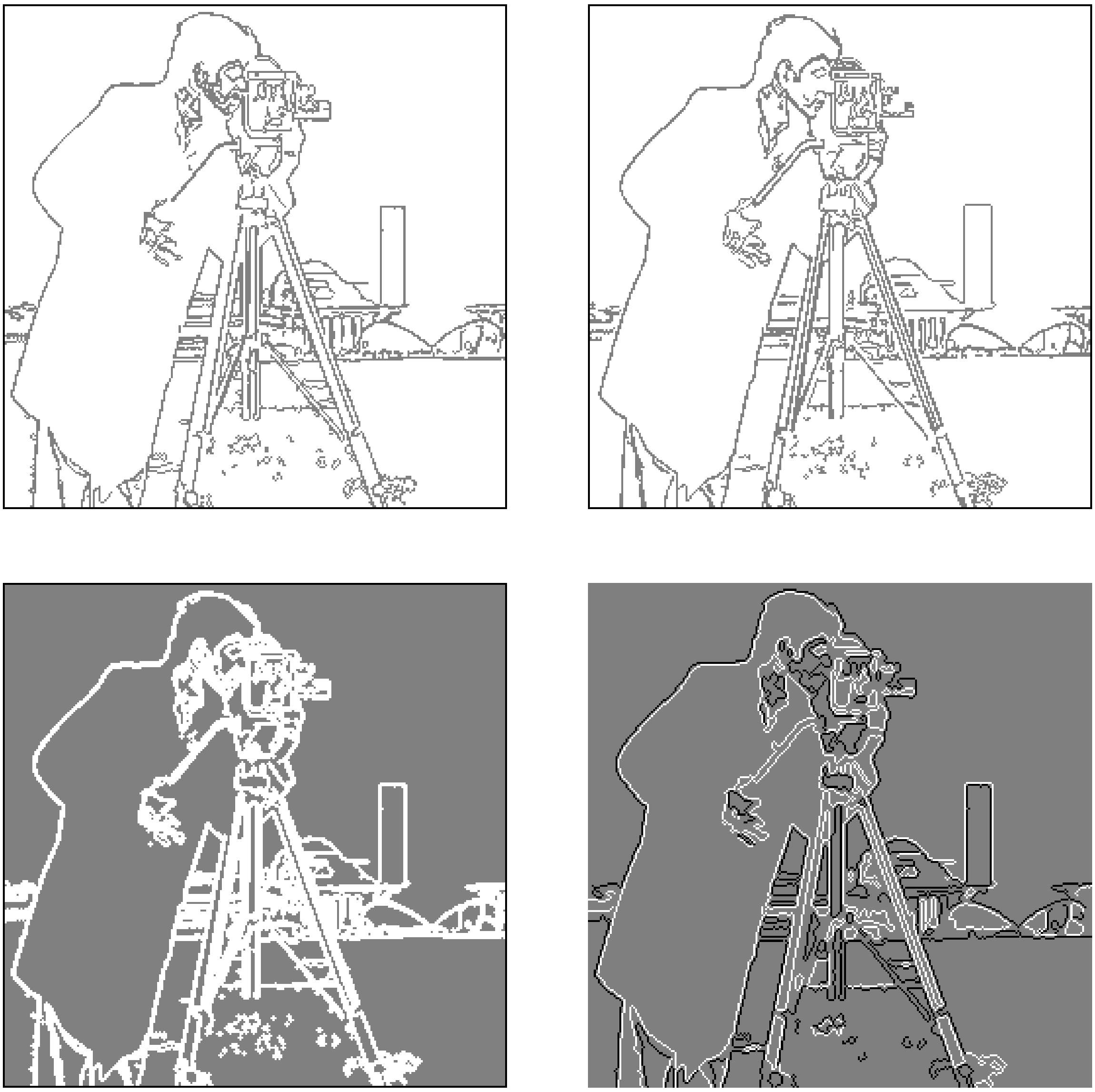


Figure 3.15: Extraction of lower and upper edge brims. Upper left: markers of negative Laplacian components. Upper right: markers of positive Laplacian components. Lower left: markers of flat areas of the image. Lower right: extracted edge brims

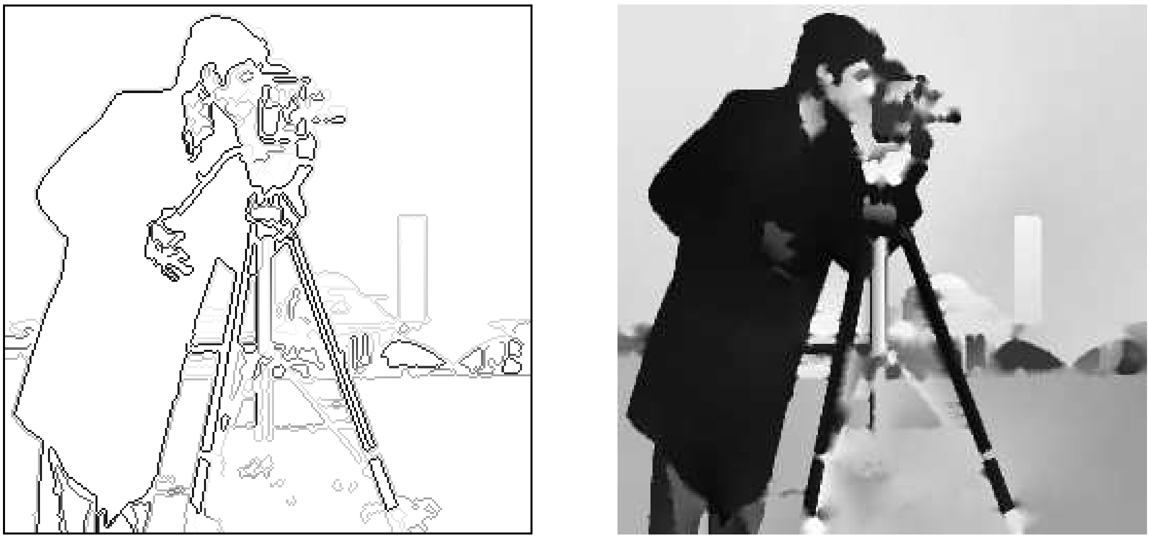


Figure 3.16: Interpolation from lower and upper edge brims: left: initial set; right, interpolation result at 0.17 bpp

Table 3.4: Compression rates for the example of Fig. 3.16

Type of information	bits per pixel	coding technique
amplitude	0.024 bpp (14%)	polynomial approx.
shape	0.136 bpp (80%)	chain-code
starting points	0.015 bpp (6%)	Elias coding
TOTAL	0.169 bpp (100%)	—

3.5 Discussion

In this chapter interpolative coding techniques have been approached from the point of view of perceptual coding by means of sketch-oriented image models. Sketch-based coding schemes are faced with two major problems: the selection of the sketch data and the interpolation process for the reconstruction of the image from such initial set. Two techniques based on morphological operators have been presented intended to solve these problems: the morphological interpolation method and the extraction of sketch data from a Laplacian image using the watershed algorithm. Up to the author's knowledge, both solutions are computationally more efficient than the ones reported in the literature[6]. The morphological interpolation technique performs a faithful reconstruction of the smooth areas from sketch data. The proposed method benefits from the properties of geometric interpolation methods (inverse distance weighting) and can be applied to any configuration of the initial set (as diffusion processes) without the large computational load of such methods. The extraction of sketch features by means of the watershed algorithm is more efficient and robust than contour tracing methods for edge extraction [12], [86].

The interpolation result of figure 3.16 corresponds to the *strong edge* component of the perceptual model that has been proposed in chapter 2. It consists of the strong edges and smooth areas of the image generated by interpolation from the positions and amplitudes of the pixels of the initial set, i.e. the lower and upper brims of strong edges. The residue of this component contains fine textures and small details. The techniques described in the following chapters are proposed for the coding of this residue.

[6] Actually, the whole process of feature extraction and coding for the image of Fig. 3.16 takes 15.3 seconds of CPU in a SPARC10. The CPU time measured only for the interpolation by linear diffusion from the same initial set (using multi-grid techniques to speed up convergence) is of 6 min. 13 sec.

Chapter 4

Coding of image details

In very low bit-rate image and video coding schemes, small visual features –*details*– are usually lost in the coding process because they are supposed to be the least significant information from the observer point of view. Occasionally, such details may be of great importance for the subjective judgment of the coded images. In video-telephone sequences, for instance, small facial shapes and shadows are specially significant even being smaller or dimmer than other features of the image. However, if all the details were to be coded, their cost in bits would be rather high. Let us take, for example, transform coding techniques. The coding of small details would imply the accurate transmission of high frequency coefficients in many image blocks. If the example taken is a segmentation-based coding scheme, without an accurate *selection procedure*, the segmentation process would probably yield over-segmented images in order to properly represent all the small details. Nevertheless, a number of meaningful details should be coded –even at very low bit-rates– in order to match their perceptual significance for the visual system [18]. Failure to code these details would certainly affect the perceived quality of the reconstructed images.

Three aspects of the problem of coding small image details will be discussed in this chapter: detail extraction, detail selection and detail coding. All these aspects are critical issues, but the key point is the *perceptual* selection step. Detail selection must be performed according to perceptual criteria, so that the system should be able to find which details are most significant to the visual perception of a human observer. Two approaches are possible from this point of view: one is the design of a knowledge-based coding system, intended to find specific image details that will be important in particular coding applications, for example, the mouth or the eyes of the speaker in video conference sequences. A second approach [14], not so application dependent as the previous one, consists in the design of a complex perceptual criterion as a combination of visual parameters of the detail such as size, contrast, etc. This criterion may be used to mark the details that would be more *visible* for the observer at the lower (physical) perception level regardless of their meaning at a higher (recognition) level. It is known, for instance, that a simple blob can be better perceived in a flat area of the image than when it is located over a highly textured background. The texture of the neighboring area of the detail, as well as any other objective measure derived from perceptual considerations, is thus a good candidate to be used as a parameter of the perceptual criterion. This second approach is the one that will be investigated in the present work.

In the paper by Jayant et al [46], perceptual coding has been stated as an imperative issue for the design of high compression algorithms. They put forward the need to *"minimize perceptual meaningful measures of signal distortion [...] for realizing high quality at low bit rates"*. Such measures should consider factors such as perceptual masking effects in order to find the exact level of just-noticeable distortion that corresponds to perfect subjective quality at the lowest possible bit-rate. In the very low bit-rate coding framework, lossless (or perceptually lossless) coding is not the target issue. In practice, for most of the input still images and video sequences, lossy coding becomes necessary in order to keep the coder output at the desired bit-rate. Lossy coding is perfectly acceptable if the information to be discarded is of little visual significance, i. e. of a rather 'indistinct' aspect. Defining the degree of visual significance of each image detail –from a perceptual point of view– would be of great help for the coding system to select the details that should be coded and the ones that should be discarded for coding.

In this work, instead of using perceptual measures of distortion, the extracted details are ranked according to several *explicit* measures of their perceptual significance. In a later step, some of the perceptually most significant details will be straightforward selected for coding. The empirical sense of such ranking is emphasized, but also its usefulness for the design of high compression algorithms, where the ultimate decision about which components of the visual content of the image must be coded depending on the available bit-rate, may be better made with the help of perceptual measures.

This chapter is organized as follows. Section 4.1 discusses why small image details have been selected as the target for this work and proposes a morphological method for the extraction of such features from images and video sequences. In section 4.2, the use of perceptual criteria for detail selection is explained. Section 4.3 describes the coding techniques that have been used for the selected details. Finally, section 4.4 presents an application of the detail coding algorithm in a segmentation-based very-low bit-rate video coding framework. Chapter 6 will present the detail coding technique in the unified framework of the perceptual model defined in chapter 2.

4.1 Detail extraction

The problem of extracting small image details from images and video sequences does not have an obvious solution. If the discussion is placed in the framework of object-oriented coding techniques performing image segmentation, it will turn out that small significant regions are not easy to obtain. A segmentation algorithm will make use of certain homogeneity criteria in order to group neighboring pixels of the image into segments or regions where the parameters defining such criteria have uniform values [82]. These parameters often rely in statistical measures performed over the amplitudes of the pixels of the region. When the size of the region decreases, the number of pixels contributing to the measure of the parameters is small and, therefore, the resulting measure is less reliable. Then, it is difficult to distinguish between true visual details –where the measure is homogeneous and different from the neighboring pixels– and non-stationarities of the local texture, that occur rather often inside textured areas of natural images. Segmentation algorithms keep a weak balance between the situation where most small details do not result in individual regions, and over-segmentation results where in addition to true details many regions are broken due to small deviations of the homogeneity measure.

In order to extract significant details from digital images, a technique strongly related to the physical image structure is required. Such a technique should deal with the shapes contained in the video signal rather than with the signal statistics for a better matching of the visual perception process. Mathematical morphology [104] provides tools that give a good insight into the structure of the images.

A morphological operator based on the top-hat transform [67] is able to find contrasted details and to extract these details successfully. This operator has been applied for coding purposes both to still images [18] and to the frames of a video sequence in intra-frame mode. Then, its use has been extended to deal with moving details in inter-frame mode [14]. A brief overview of the basic morphological operators that will be used in the sequel for the definition of the detail extraction technique can be found in Appendix A.

4.1.1 Morphological operators for detail extraction

The morphological top-hat transform is defined as the difference between the identity operator and the morphological opening or, in the dual case, between the closing and the identity. They extract from the original image bright or dark contrasted components smaller than the structuring element. However, the top-hat also extracts spurious components from the contours of larger objects that have been modified by the morphological opening or closing. To avoid the extraction of spurious components, one may also choose the reconstruction top-hat, computed from an opening or closing by reconstruction.

Let us call x_i the values of the original image on the points i of the definition space. The morphological top-hat tht_i and the dual (black) top-hat $btht_i$ are defined as follows:

$$\begin{aligned} tht_i &= x_i - \gamma_n(x_i) \\ btht_i &= \varphi_n(x_i) - x_i \end{aligned} \tag{4.1}$$

whereas the reconstruction top-hat $tht_i^{(rec)}$ and the dual (black) reconstruction top-hat $btht_i^{(rec)}$ are given by the following expressions:

$$\begin{aligned} tht_i^{(rec)} &= x_i - \gamma^{(rec)}\left(\varepsilon_n(x_i), x_i\right) \\ btht_i^{(rec)} &= \varphi^{(rec)}\left(\delta_n(x_i), x_i\right) - x_i \end{aligned} \tag{4.2}$$

A synthetic image composed of geometric elements and one frame of the original sequence *car-phone* are shown in Fig. 4.1. The synthetic image will serve to better identify the effects of the morphological operators in the natural one. Figs. 4.2 and 4.3 respectively present the smoothing characteristics of the morphological open-close filters $\varphi\gamma$ and the open-close by reconstruction $\varphi^{(rec)}\gamma^{(rec)}$ and the corresponding top-hat operators $Id - \varphi\gamma$ and $Id - \varphi^{(rec)}\gamma^{(rec)}$. Notice the changes produced by morphological open-close filters along the contours of the large objects that remain after the filtering stage in the left images of Fig. 4.2. This produces spurious components in the morphological top-hat that do not correspond exactly to 'objects' in the image. Rather they are part of larger objects, as the small extensions of the contours in the synthetic image or the seams and the shoulders of the jacket in the natural one. These spurious components may not be considered 'true' image details. If the reconstruction top-hat is used, the contour shapes of large objects are preserved, but some shapes of the extracted details remain sometimes visible on the filtered image after the reconstruction process. They simply get the same grey value as the neighboring objects, becoming an extension of them. This effect is clear in the filtered images of Fig. 4.3. There has been an incomplete extraction process for the details of the synthetic example and, for instance, the tree that can be seen through the window of the car gets the amplitude level of the bushes in the background.

In order to overcome the drawbacks of both top-hat operators, a new morphological transform has been proposed by Meyer in [68]. It is based on the detection of 'true' details from the reconstruction top-hat and the computation of its real amplitude values from the morphological top-hat. A marker image indicates the position of the pixels of the reconstruction top-hat whose amplitudes are over a certain contrast threshold λ. This image is called the

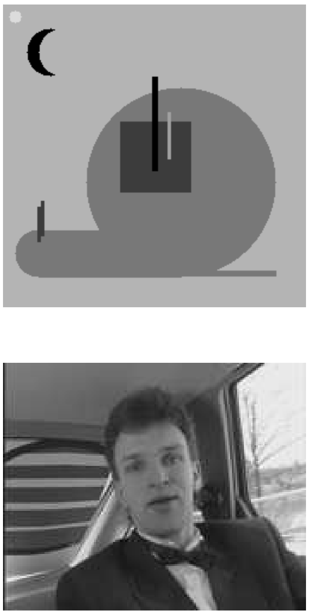


Figure 4.1: A synthetic image and one original frame of the *car-phone* sequence

marker of bright details mkr_i and is defined as follows:

$$mkr_i = \begin{cases} 255 & \text{if} \quad tht_i^{(rec)} > \lambda \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

The image of bright details $detw_i$ is then obtained by geodesic reconstruction of this marker image under the morphological top-hat:

$$detw_i = \gamma^{(rec)}(mkr_i, tht_i) \tag{4.4}$$

The image of dark details $detb_i$ is obtained in a similar way means of the dual operators. Finally, the difference of $detw_i$ and $detb_i$ results in an image containing both bright and dark details, called det_i. The extracted details are thus:

$$det_i = detw_i - detb_i \tag{4.5}$$

and the smoothed image smt_i is then computed by subtracting det_i from the original:

$$det_i = detw_i - detb_i \tag{4.6}$$

The use of the reconstruction top-hat to obtain the marker images guarantees, on the one hand, that the artifacts due to contour smoothing –not present in the reconstruction top-hat–

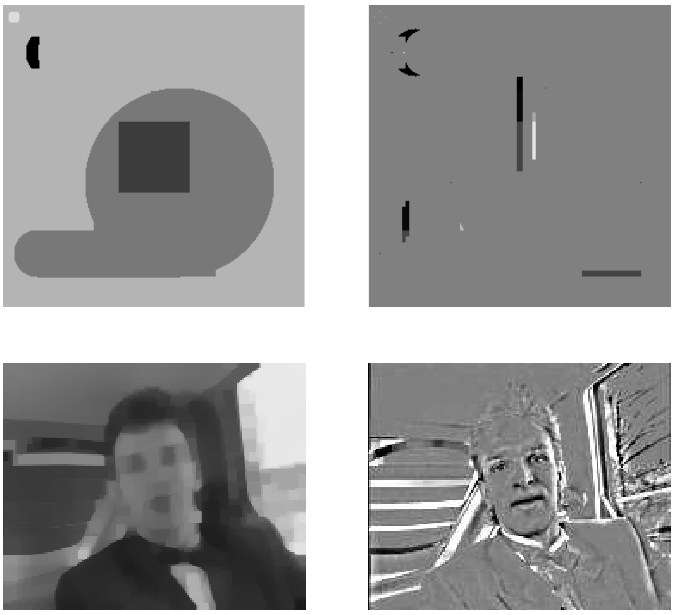


Figure 4.2: Morphological open-close of size 2 (left) and morphological top-hat: $Id - \varphi\gamma$ (right) of the images in Fig. 4.1

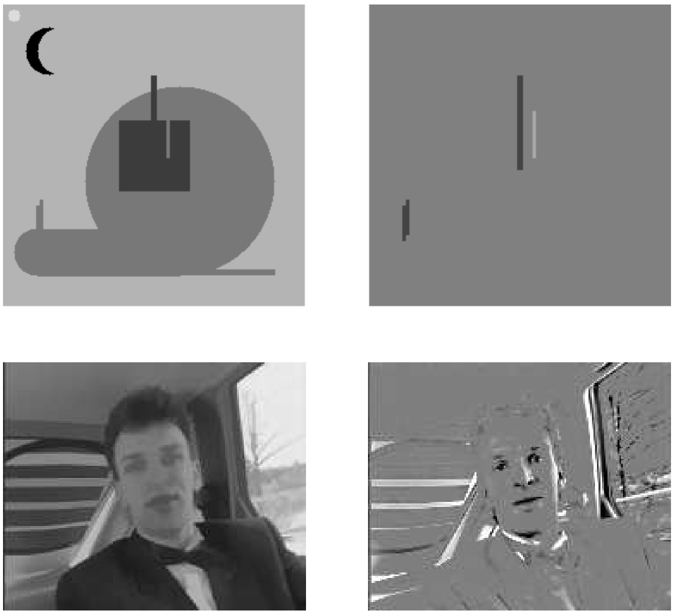


Figure 4.3: Open-close by reconstruction of size 2 (left) and reconstruction top-hat: $Id - \varphi(rec)\gamma(rec)$ (right) of the images in Fig. 4.1

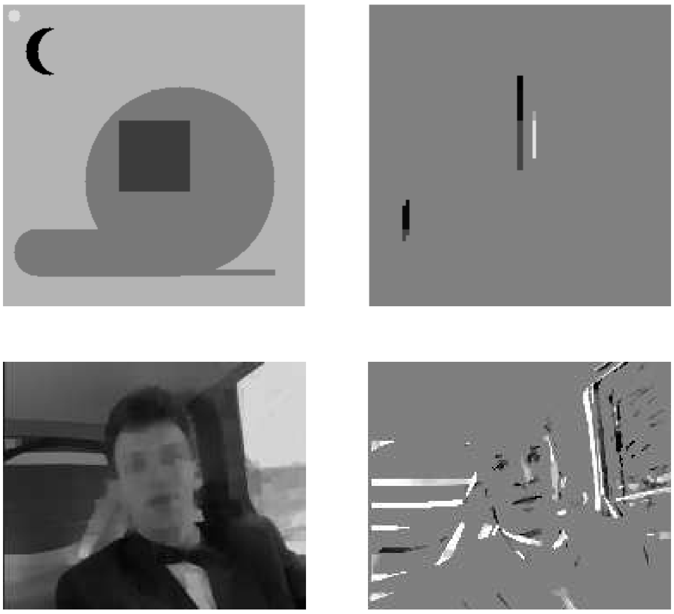


Figure 4.4: Smoothed image *smt* after detail extraction (left) and extracted details *det* (right)

will not be reconstructed and, on the other hand, that the extracted details will get the true amplitude values from the morphological top-hat. Fig. 4.4 illustrates the application of this detail extraction operator to the original images of Fig. 4.1. It is worthwhile to observe that it performs the detail extraction *perceptually*, in the sense that the extracted components approximately correspond to perceived visual features. There are not spurious components due to contour smoothing and the extracted details disappear completely from the smoothed images. Moreover, in the smoothed images the locations of these details are filled up with the amplitudes of the neighboring pixels, so that they seem to be replaced by the (intuitive) underlying background.

4.1.2 Detail extraction from video sequences

In order to deal with moving details in video sequences, we propose the extension to the temporal dimension of the technique for still images described in the previous section. The details of the first frame of the sequence are obtained as explained above. Then, the extraction algorithm follows the temporal changes of the extracted details in the forthcoming frames in inter-frame mode.

As illustrated in Fig. 4.5, in inter-frame mode some new details appear in each new frame, some details disappear and most of them are kept but may vary their attributes of shape, position and amplitude. The extraction process should track the preserved details in the current frame and identify the new ones. To this end, the details obtained in frame $t-1$ are used as additional markers in frame t for the geodesic reconstruction process defined in eq. 4.4. Let us assume that the extracted bright details in the previous frame are denoted by $detw_{i,t-1}$. The marker for bright details in frame t is in this case:

$$mkr_{i,t} = \begin{cases} 255 & \text{if} \quad tht_{i,t}^{(rec)} > \lambda \quad \text{or} \quad detw_{i,t-1} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.7}$$

To obtain the bright details in frame t of the video sequence the marker is reconstructed under the top-hat as in eq. 4.4 (dark details will be obtained, as usual, by the dual morphological operators):

$$detw_{i,t} = \gamma^{(rec)}(mkr_{i,t}, tht_{i,t}) \tag{4.8}$$

The target of this marking is twofold: on the one hand, it allows to follow each particular detail through the temporal dimension by means of an additional sequence of label images. This labeling will be explained in more detail in the sequel. On the other hand, the marking contributes to the temporal stability of the extraction process, because it forces the continuation of the old details in the current frame. Of course, if any of these details is not present any more, its reconstruction will not be possible because it will not appear in the reference image $tht_{i,t}$. The marking is necessary just in case the amplitude of an old detail goes below the threshold λ but does not disappear (simply becomes dimmer). Such detail, that would not be extracted in intra-frame mode, will be obtained now in the current frame and its temporal continuity will be kept.

Notice that, in the previous discussion, one important assumption has been made about the temporal connectivity of details: two bright (or dark) details extracted from consecutive frames are supposed to be the same if they are 'connected' through the temporal dimension. That is, if both details were projected into the same temporal plane, let us say in $t = 0$, they would have at least one pixel in common. This temporal connectivity criterion is rather simple, but it is efficient to decide whether a particular detail continues in consecutive frames. Nevertheless, it prevents fast moving small details from being connected through consecutive frames and, thus, from being extracted as only one feature across the time dimension. The minimum speed in pixels per frame at which a moving detail will be disconnected is precisely its width in the direction of the movement. Actually, the extraction of fast moving details

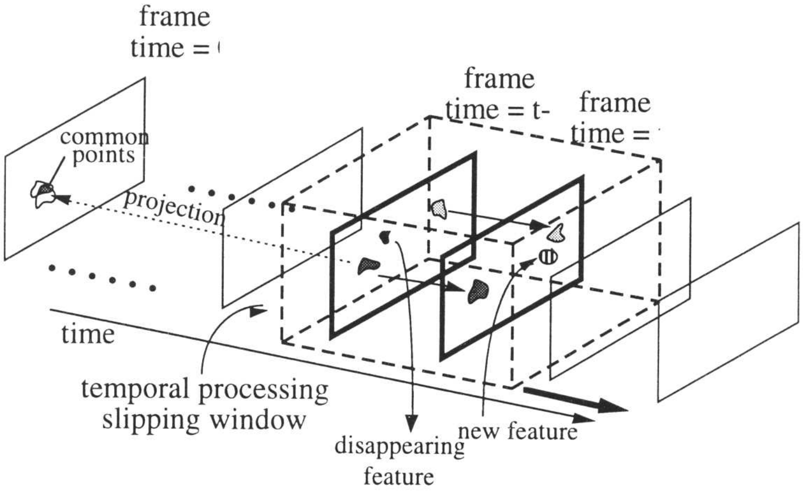


Figure 4.5: Illustration of the temporal variation of moving video details

will not be a great problem, because they will be extracted as separate new details if they are significant enough. Motion information obtained from the evolution of the details in previous frames could be used to connect such details at the expense of a considerable increase of the complexity of the extraction method.

A second possible problem regarding temporal connectivity comes up when one detail is split into two or more spatially disconnected components in the new frame. Then a decision has to be made in order to choose which one will be taken as the continuation of the same detail in the current frame and which one will be assumed to be a new detail. A simple similarity criterion such as the number of common points of the projections in $t = 0$ of these components and the detail in the previous frame can be used in this case.

4.2 Perceptual ranking and selection of details

In most natural images, the number of details extracted in each frame is often large. However, not all the extracted details are of equal importance in terms of their contribution to the perceived image quality. As pointed out in the introduction, the coding scheme must be able to identify and code those details that are the most significant to the human eye. Therefore, the detail selection step should be performed according to perceptual criteria. The available bit-rate at the output of the coder sets a severe limit on how many details can be coded by the system. In very low bit-rate video coding, the experience shows that selection ratios of one detail out of ten are the common situation. The highest the compression, the most important the selection step for the overall performance of the coding scheme. For instance, if it is possible the coding of, let us say, only twelve details per frame, a powerful and robust detail selection algorithm will be necessary in order to decide which twelve details among the extracted one hundred and twenty are the most significant for the visual system. This decision should be taken at each frame. The accuracy of the selection step becomes then of capital importance for the visual quality of the reconstructed images.

The selection of visual details plays a decisive role in any object-oriented coding scheme. Most systems perform this selection in an implicit way, by tuning certain parameters that directly affect the feature extraction step. In segmentation-based coding, for example, if the homogeneity criterion is relaxed, some regions are merged into larger ones, so that the decision contours separating the 'objects' disappear and do not need to be coded, what implies that the different objects are considered the same. In the present work, however, an *explicit perceptual criterion* is used for the detail selection step.

4.2.1 Design of an explicit perceptual criterion for the selection step

The design of the selection step relies on the answer to the following question:

- What objective parameters make some details most *visible* for the observer?

Candidates for the answer are, of course, the size and the contrast of the detail. Details are meaningful if they show some significant contrast over the background, even if they are small. Details showing low contrast levels will be *visible* only if they have some significant size. A possible measure of the interaction of both parameters may be given by what could be called the *energy* of the detail, defined as the product of the average contrast by the size. On the other hand, the masking effect of the surrounding background should also be considered [21]. Details located near sharp edges of the image will be masked by the strong visual effects of such edges. The texture of the background will also affect the perception of the detail: tiny details over smooth areas may be easily perceived, whereas a stronger visual stimulus is necessary to excite the visual perception over a highly textured background. Furthermore, the perception of the detail is affected by the luminance of the background according to Weber's law[71]. More parameters of visual details may be included in this list. Interesting ones are the dynamics [41] of the details, considered as image extrema, and the average distance to significant features in the neighborhood [21]. Both measures give and idea of the global importance of the detail with respect to other features in the image. The dynamics is able to distinguish regional extrema from local extrema of the image, whereas large values for the average distance indicate that the detail is important because there are not significant features nearby that could mask its perception.

Up to now, only spatial parameters of still image details have been discussed for perceptual considerations. When dealing with video sequences, temporal measures are of great help to assess the perceptual significance of moving details. Details presenting a certain degree of persistence through a certain number of frames are, of course, much more likely to be significant than those details appearing only for a short period of time. The temporal behavior of details would be an important parameter to be considered for the detail selection step.

4.2.2 Empirical formula for detail selection

In practice, parameters like those proposed above should merge in a complex perceptual criterion including also mutual interdependences and masking effects [66]. With this criterion, the coding scheme would try to imitate the performance of the human visual system in the perceptual importance given to each detail. As the mechanism of the visual perception is very complex and by no means completely known, only rough approximations to this target may be attempted. The aim of this section is to prove the usefulness of such approximation for the selection step in critical situations where severe selection of the information to be coded must be carried out.

The extracted details are first labeled in order to take independent perceptual measures for each one. The size of a given detail of label k in the current frame will be the number of pixels affected by label k in the label image. The contrast of the same detail can be measured as the average value of the pixels in the detail image $det_{i,t}$ under the respective label. Measures of texture activity or proximity of strong edges may be considered in the neighborhood of each label by, for instance, taking the variance of the gradient under a dilated version of the labels. Moreover, if the labeling is consistent with the temporal connection of the details between consecutive frames, it is possible to keep a record of the evolution of the parameters of each detail along the time. The cumulative measures stored in such variables from preceding frames would serve to assess the temporal behavior of each detail.

Several tests have been made with empirical combinations of perceptual parameters. A practical measure that gives good subjective selection results is the one described in the sequel. Being $lab_{i,t}$ the sequence of label images for the extracted details at frame t, the following parameters may be defined in the current frame:

- size parameter for detail k:

$$siz_{k,t} = count_i(lab_{i,t} = k) \tag{4.9}$$

- contrast parameter for detail k:

$$con_{k,t} = mean_i(det_{i,t}, i : lab_{i,t} = k) \tag{4.10}$$

- energy parameter for detail k:

$$\mathcal{E}_{k,t} = siz_{k,t} \cdot con_{k,t} \tag{4.11}$$

- activity of the surrounding background:

$$\mathcal{A}_{k,t} = variance_i\,(grad_{i,t}, i : \delta_n(lab_{i,t}) = k) \tag{4.12}$$

The previous parameters are combined in the following expression in order to obtain an empirical measure of perceptual significance for the detail k in frame t:

- empirical measure of perceptual significance

$$rank_{k,t} = \frac{\mathcal{E}_{k,t}}{\mathcal{A}_{k,t}} + \beta \cdot rank_{k,t-1} \tag{4.13}$$

The constant β takes values in the interval $[0, 1]$ and is used to tune the balance between the spatial parameters obtained for each detail in the current frame and the temporal evolution of these parameters in previous frames. Values of β close to one will increase the relative importance of the temporal persistence of the detail, whereas for the value $\beta = 0$, only the spatial parameters measured in the current frame are used to assess the visual importance of the detail. In intra-frame mode β is set to 0, whereas in inter-frame mode the best results have been obtained for values of β close to 1.

4.2.3 Actual selection of details

Once the empirical measure of perceptual significance $rank_{k,t}$ has been estimated for the extracted details of the current image, the problem of selecting the most significant ones is already solved. Details resulting with the highest values for such empirical measure are selected straightforward. If the available bit-rate for detail coding is high, more details will be selected for coding, but if only a small number of them can be coded, the ability of the

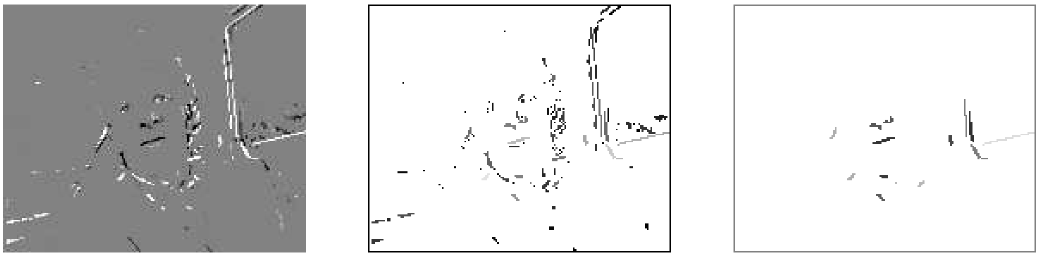


Figure 4.6: 150 extracted details (left), image of rank values (center) and 19 selected details (right)

empirical ranking of details to imitate the performance of the visual perception will be really put to test.

An example illustrating the selection from the rank values obtained with the empirical formula of eq. 4.13 is shown in Fig. 4.6. The rank values are represented as grey level values of the detail labels in the center image. The higher the rank, the brighter the label appears in this image. Notice that, except for the subjective meaning of some of the details due to a higher level of recognition, the ranking is not very far from the result of a manual classification of the extracted details performed by a human observer according to their ‘visual’ importance.

4.3 Coding of selected details

The coding of small features in still images and small visual moving components in video sequences is a rather troublesome task for any video coding scheme. That is the reason why most very low bit-rate coders skip the problem of coding small details or simply encode some rough approximation of them. In this section a proposal for the efficient coding of the selected details is presented. The coding strategy may sometimes benefit from psycho-visual properties in order to increase the efficiency without significant quality degradation. Three types of data are coded for the temporal section of each selected image detail in a given frame:

- amplitude values,
- position
- and shape.

The exact amplitude levels of small details are not very accurately perceived by the visual system, because they are mainly formed by high frequency components and the sensibility of the eye to these components is not very high [21]. Therefore, the amplitude levels Y, R-Y and B-Y of the signals defining the color of the detail are PCM coded with a few bits in the first frame where the detail appears. In the following frames, DPCM coding of these values is performed: the coded levels are updated by computing and coding only significant variations of the amplitude values of the detail in the current frame with respect to the previous coded value. Finally, the coded amplitudes of the extracted details are stored in a buffer in the order of the details labels and an arithmetic coder is applied for the entropy coding of the symbols of this buffer.

The bulk of the coding effort, however, is devoted to the shape and position information of image details. In intra-frame mode, contour coding techniques like chain-code [34] may be used to encode spatial information of detail labels. It has been found that chain-code is not very efficient when details are smaller enough to have more contour pixels than inside (texture) pixels. In addition, as detail contours are usually not connected among them, the coding of the coordinates of the initial points for each contour severely penalties chain-code techniques. Furthermore, the extension of chain-code to inter-frame mode is not straightforward.

The coding of this type of spatial information could be performed very efficiently by means of run-length coding techniques. In particular, a modification of the relative element address designate coder (READ) described in [130] allowing the extension of multi-dimensional run-length coding to the temporal dimension is proposed. In intra-frame mode, READ coding of small image details has been proved to be 20% more efficient than chain-code. READ coding consists in coding the runs in the first line of the image and, in the following lines, the differential runs with respect to the transition positions in the preceding line (reference line) are encoded. A detailed explanation of this technique can be found in [130].

However, it is in inter-frame mode, where the proposed modification of READ coding shows its utility. The displacements of detail labels between two frames are estimated to perform motion compensation of the details before coding. Then, once motion compensation is applied, the READ coder has several reference lines available in order to compute differential runs for the coding of the current line:

- the previous line in the scanning order, as usual
- and the closest lines from the previous frame.

Fig. 4.7 gives an illustration of these possibilities. The reference line that produces the shorter codes for the differential lengths is chosen and one new symbol indicating which line has been selected as reference is introduced in the output buffer. In practice, the coder results in

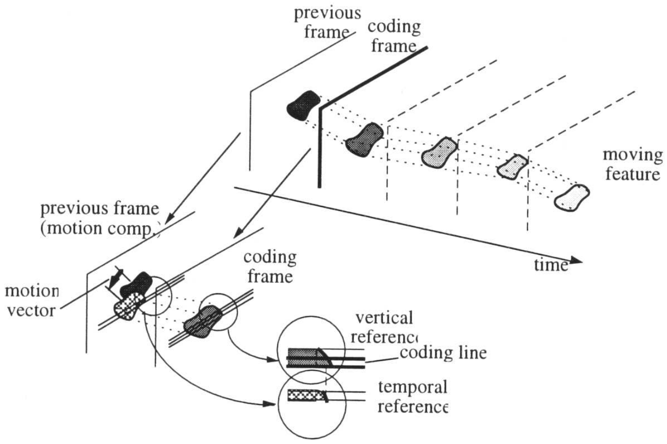


Figure 4.7: Motion compensated adaptive READ coding for image details

an adaptive READ method with several *pass modes* and *temporal modes* in addition to the common vertical, horizontal and pass modes.

4.4 Examples of application

The detail extraction, selection and coding technique that has been presented in the previous sections may be used as the complement of an object-oriented coding scheme for image sequences. A segmentation-based coding scheme will be used as the basic system to illustrate the coding of details.

The reference coding system is based on a three dimensional morphological segmentation algorithm. It was originally developed for still image coding [96] and later extended to video sequences [94]. A contour-texture approach is used for the coding of the regions resulting in the segmented sequence. In particular, it is able to produce coded sequences of different qualities at different bit-rates.

Fig. 4.8 shows two sample coded frames for the *car-phone* sequence. Only the luminance component is shown, but the bit-rates include both the luminance and chrominance components for a frame rate of 5 Hz, which is often used in very low bit-rate video coding applications. A cost-efficient texture coding technique has been used for the interior of the regions [37]. It approximates textures of the regions by a weighted sum of orthogonal

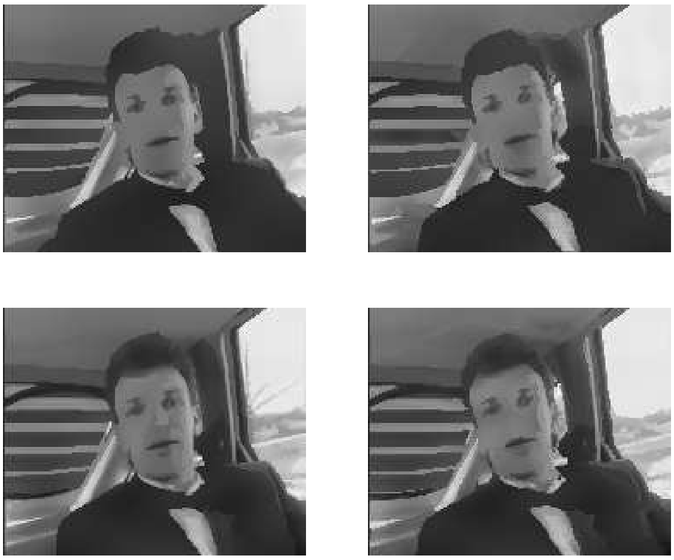


Figure 4.8: Two coded frames of the sequence *car-phone* at 21 Kbit/s (top row) and 40 Kbit/s (bottom row)

cosenoidal basis functions of low order. The sequence in the top row has been coded at 21 kbit/s: 14 Kbit/s for contour information, 6 Kbit/s for textures and 1 Kbit/s for motion. The sequence in the bottom row has been coded at 40 Kbit/s: 28 Kbit/s for contour information, 9.5 Kbit/s for texture and 2.5 Kbit/s for motion. The rate of 30 Kbit/s has been found to be the minimum necessary bit-rate for the system to encode significant details such as those appearing in the face of the man. The perceptual quality improvement at this bit-rate is significant, but the cost has been rather high.

The detail extraction, selection and coding scheme explained in this chapter has been used to improve the perceived quality of the segmentation at a lower cost. The system carefully analyzes the perceptual significance of each one of the new details that may be selected for coding. This results in a high coding rendition because only the most meaningful details from the perceptual point of view are considered. Fig. 4.9 shows the improved results obtained by adding an average number of 20 details to the segmentation result of the top row of Fig. 4.8. The details shown in Fig. 4.9 were coded at 4.2 Kbit/s. Table 4.1 presents the distribution of this rate among the different types of coded information.

The reconstructed image of Fig. 4.9 has been coded using a total bit-rate of $21 + 4.2 =$

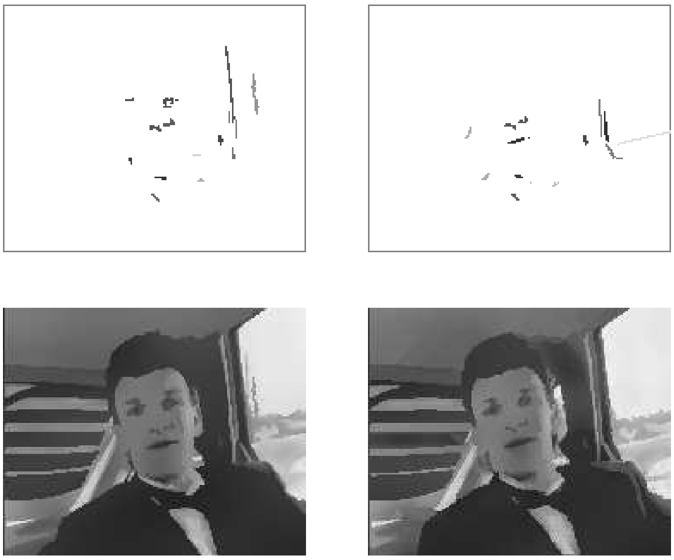


Figure 4.9: Some selected details (top) included in the coded segmentation shown in the first row of Fig. 4.8 result in a subjectively improved reconstruction at 25.2 kbit/s (bottom)

Table 4.1: Rate figures for the 20 coded details

DATA TYPE	rate (kbit/s)
position (READ)	2.9
motion vectors	0.4
amplitudes	0.9
details rate	4.2
total rate (whole sequence)	25.2

25.2 kbit/s instead of the 40 kbit/s required for the coding result of the bottom row of Fig. 4.8. From these results it can be seen that the perceptual selection of image details is able to increase significantly the rendition of the coding scheme. A different strategy for quality improvement may be to employ a more complex texture coding technique in order to improve the rendition of the regions' interiors. Observe that this could be efficient from the point of view of coding but presents two disadvantages: first, it would give the same importance to all the small features that are missing in the whole image and, second, it would introduce 'waveform-like' coding artifacts in the texture of the regions. The identification of the individual details guarantees the selection and coding of each one of the missing image features according to a complex measure of its visual significance.

Discussion

The novelty of the coding strategy presented in this chapter is precisely the study of the visual significance of image details by means of explicit perceptual measures obtained from objective, local properties of each individual feature. The results prove that such measures deserve consideration in advanced coding schemes based perceptual image models. Meaningful image details may improve the subjective quality of the reconstructed images at a minimum cost.

Morphological operators are useful shape-oriented analysis tools that can be used in the spatial-temporal domain for detail extraction. Once extracted, a perceptual selection is performed in order to keep only the most significant details. An efficient coding technique has been proposed for the coding of these details, which is based on motion compensation and relative addressing run-length coding. Results using the proposed technique in a more general three component model will be presented in chapter 6.

Chapter 5

Region–based subband coding

Subband coding is based on the decomposition of the input image into frequency bands. Each band is decimated and coded separately, using a quantizer and a bit-rate accurately matched to the statistics and visual importance of that band. The use of *quadrature mirror filters* (QMF) in the analysis/synthesis stages makes possible an alias-free reconstruction of the original signal.

The main advantage of subband coding schemes is that the quantization noise generated in a particular band is largely limited to that band in the reconstruction, not being allowed to spread to other bands. Moreover, by varying the bit assignment among the subbands, the noise spectrum can be shaped according to the subjective perception of noise by the Human Visual System[46]. This leads naturally to a pleasing image reconstruction from the point of view of perceptual image compression. In addition, the subband decomposition allows straightforward progressive multi-resolution transmission.

The first use of subband analysis for image coding is often attributed to Schreiber [102], who reported in 1959 the system known as *Synthetic Highs*, but Schreiber himself [101] cites Kretzmer [53] as the first to use subband coding for television signals. They showed that fewer bits per sample could be used for the higher- than for the lower-frequency bands in most natural images. A similar scheme, the *Laplacian pyramid*, was presented by Burt and Adelson in 1983 [11]. Pyramidal coding was introduced as a non-causal predictive coding scheme, where a low-pass prediction of each pixel is obtained as a local weighted average based on a symmetric neighborhood centered at the pixel itself. Then, the prediction error image containing high-pass frequency information was quantized and entropy coded and the same decomposition procedure recursively applied on the down-sampled low-pass image.

Subband schemes obtained greater data compression than sequential (causal) prediction techniques and were simpler to implement than transform techniques. However, the concept of quadrature mirror filtering was not applied to image signals until the theoretical extension of one-dimensional (1-D) QMF filtering to multi-dimensional signals was treated by Vetterli [118]. In particular, the application of subband coding to images by means of two-dimensional (2-D) QMF separable filter banks was introduced by Woods and O'Neil in [128]. The advantages of 2-D QMF filtering techniques to the subband image coding approach are based on:

1. Subsampling the high frequency (or prediction error) images is possible in order to obtain a critically sampled decomposition[1]. This reduces by a factor of 3/4 the total number of samples before quantization and coding with respect to the Laplacian pyramid scheme.

2. The frequency selectivity characteristics of the filter bank allows the extraction of the spatially oriented structural redundancy typically found in natural images.

These two facts definitely improved the compression ratios achieved by the pyramidal scheme, offering superior coding performance to that of the early subband systems and bringing forth new possibilities for directional decomposition and edge-oriented perceptual coding systems as the one already presented by Kunt et al in [54].

Nowadays, subband coding [127] is a powerful method of image and video compression, able to compete successfully with the well-established block transform methods which have been the state of the art for the past two decades. Subband coding does not produce the blocking artifacts that arise when block processing is performed in high compression transform coders. Unfortunately, the human eye is very sensitive to this type of distortion and, therefore, block coders are not appropriate for low bit-rate image coding. However, at low bit-rates subband coding presents a distinct type of distortion due to the Gibbs phenomenon of linear filters. This distortion, called 'ringing effect', is visible around high-contrast contours and can also be very annoying. Although it is possible to reduce the ringing effect by an appropriate design of the subband filters [109], [27], it is not possible to find linear subband filters without any ringing effect.

To avoid ringing artifacts, morphological filters [132], [93] can be used. Morphological multi-resolution analysis decomposes the image into different filtered images, each containing objects, for instance, of a specific size or a specific contrast. The filtered images are obtained

[1] A filter bank is said to be *critically sampled* if the total number of samples in the subband signals is equal to the total number of samples in the original signal [23]

by simply computing the residues of a cascade of open-closings. However, its major disadvantage for coding purposes is that, if perfect reconstruction is required, no down-sampling can be applied [43] and, hence, all the filtered images are of the same size as the original image. In *morphological pyramidal schemes* [113], [111], down-sampling is performed after each open-closing, but the interpolation error has to be fed back into the 'high-resolution' filtered images in order to obtain a lossless decomposition. Nevertheless, these schemes cannot compete with linear subband coding of images where critical sampling is performed.

Several proposals have been made of image decompositions using critically sampled morphological filter banks [78], [79], [31], even preserving the perfect reconstruction property [16], [28]. These schemes do not present any ringing effect, but the quantization on the high resolution images obtained with morphological filters yields poorly represented textured regions in comparison with their linear counterparts. To overcome such problem, an adaptive decomposition has been introduced in [28] which selects linear filters on textured regions and morphological filters otherwise.

In this chapter, we present a different approach that may be employed to avoid ringing artifacts around high contrast edges. Linear filter banks (QMF) are proposed for the subband analysis and reconstruction stages, but the filtering procedure is modified so that it can be applied inside relatively homogeneous regions, usually separated by strong edges. Pixels belonging to one region, on one side of the edge, will not be filtered together with pixels of the neighboring region located on the other side. Therefore, oscillations of the filtered image around strong edges will be less noticeable. This approach leads to *feature-based* [17] or *region-based* [55], [5] subband coding.

One of the main advantages of subband coding over transform coding techniques is that it makes possible to adapt the coding process over arbitrarily shaped objects extracted from each subband. But the advantages of applying the subband decomposition inside homogeneous regions, and the underlying perceptual model behind it, go well beyond than only solving the problem of ringing effects.

The most significant characteristics of a region-based subband coding scheme are described in section 5.1. The application of separable subband analysis filters on arbitrarily shaped regions, requires one-dimensional subband decomposition of arbitrary length signals. Section 5.2 describes how to solve the problems at the boundaries of limited extension signals (such as image regions) due to the effect of the filter when crosses the signal border (or region boundary). Next, the quantization and bit-allocation problems are discussed in section 5.3. The last section of the chapter, is devoted to some illustrative examples of region-based subband coding and its performance is compared with other texture coding approaches that have been proposed in segmentation-based compression schemes.

5.1 Advantages of region–based subband coding

Conventional waveform coding techniques aim at the extraction of the spatial redundancy present in natural images. Pixel-to-pixel correlation is very high inside homogeneous regions, but not across the edges or sharp transitions. Pixel-based coding techniques generally fail near strong edges (as conventional subband coding does) either not extracting the redundant information or showing annoying artifacts if they try to do so. This results in low compression ratios or in low image quality for high compression applications.

The special role of strong edges in our perception of images has been pointed out in chapters 1 and 2. An important function of coding algorithms is to render edge information faithfully, regardless of the fact that strong edges usually have small spatial support. Edge-based or region-based coding techniques are proposed as a solution to this challenge. Region-based subband analysis fully exploits the information about the edge structure of the images. Ringing artifacts around sharp transitions are clearly diminished compared to conventional subband coding schemes, whereas the ability of subband coding for the coding of homogeneous regions is kept. This produces clear subjective improvements which are especially noticeable at low and very-low bit-rates.

Both the characteristics of natural images and the perception properties of the human eye should be considered in the design of the coding scheme. Let us cite the most significant characteristics of region-based subband coding in this context:

- Subband analysis may be performed taking into account the proximity of strong edges. The image signal may be filtered on one side and up to the transition, either using space-variant filters or properly extending the signal values before filtering.

- Similarly, the quantizer steps can be varied depending on the proximity of the considered edges. Different sets of quantizers for the image subbands can be applied as well for regions having different contents: flat, smooth, textured or highly textured regions.

- A careful analysis of the type of information that is actually coded in the quantized subbands of a conventional linear subband decomposition reveals that it mainly corresponds to the strong transitions of the original image. The transition information is spread over several frequency bands, resulting in an important cost for the coding system and, in spite of this fact, the reconstruction of the main transitions at low bit-rates is rather poor. The separate filtering and coding of the pixels at both sides of the transition will result in a reduced contribution of edge features to the zeroth-order entropy of the higher frequency subbands after quantization, thus allowing higher compression ratios.

- However, the positions of the separating contours should be coded, so that at the decoder end the synthesis filters may perform the reciprocal subband reconstruction. Of course, the overhead contour bits will increase the bit-rate. A trade-off between small–and–highly–homogeneous regions with many contour pixels and large–and–less–homogeneous ones needing a smaller amount of contour information should be found. The cost of taking into account a given contour should be considered and a decision on a rate-distortion basis should be made in order to solve such trade-off.

- Region-based subband analysis yields a twofold decomposition: in the spatial and in the frequency domains. Rate-constrained quantizer optimization may be performed on both domains independently. The number of bits for coding are distributed over the frequency bands of each region, with the only constraint of the total available bit-rate.

Therefore, the coding of the region-based frequency decomposition can be made fully *adaptive* to the information contents of each region and, besides, also adaptive to the subjective perception of such contents by the visual system. The coder will be free to vary all the parameters involved in the analysis and coding stages at the cost of some overhead information. Namely, the type of filters and length of the filter responses, the number of recursive levels, the structure of the decomposition for each level –i.e., the width and depth of the frequency decomposition tree–, the set of quantizers applied to the various subbands of each region and, finally, the bit-allocation strategy which may be modified, by varying the measure of distortion employed, according to the perceptual importance of the region given its size, texture contents or the strength of neighboring edges.

The issues related to filter design, quantization and bit-allocation, will be treated in the following sections, whereas the discussion concerning the edge structure (or the partition) considered in the spatial domain is left to chapter 6.

5.2 Subband filters and signal extension

In general, a subband coder consists of two stages: 1) the analysis filter bank along with decimation operators, used for the filtering and down-sampling steps; and 2) a coder which encodes the subband images for the purpose of storage or transmission by means of quantization and entropy coding. At the decoder end, the reciprocal blocks are found, i.e. entropy decoder, inverse quantizer and the synthesis filter bank with interpolators for the up-sampling operation. Let us consider in this section the analysis/synthesis filter bank.

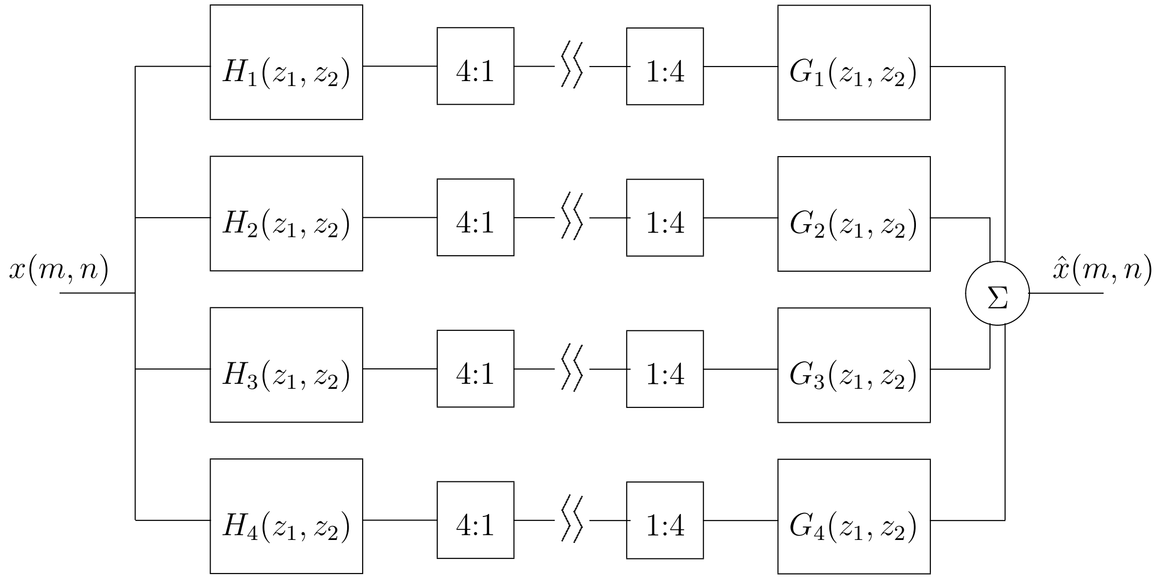


Figure 5.1: Two-dimensional subband analysis/synthesis filter bank

5.2.1 QMF filter bank

Fig. 5.1 represents a 4-channel 2-D subband analysis–synthesis system without the coding stage. Let $H_i(z_1, z_2)$ and $G_i(z_1, z_2)$ be the z-transforms of filters $h_i(m, n)$ and $g_i(m, n)$ for $i = 1, 2, 3, 4$. In a relatively straightforward manner (see appendix B, section B.3), it can be shown that the following set of conditions are sufficient for an alias-free reconstruction of the two-dimensional input signal $x(m, n)$ at the receiver side:

$$H_1(z_1, z_2) = G_1(z_1, z_2) = H(z_1, z_2) \tag{5.1}$$

$$H_2(z_1, z_2) = -G_2(z_1, z_2) = H(z_1, -z_2) \tag{5.2}$$

$$H_3(z_1, z_2) = -G_3(z_1, z_2) = H(-z_1, z_2) \tag{5.3}$$

$$H_4(z_1, z_2) = G_4(z_1, z_2) = H(-z_1, -z_2) \tag{5.4}$$

$$H(z_1, z_2) = H_{z_1}(z_1) H_{z_2}(z_2) \tag{5.5}$$

The set of two-dimensional filters $H_i(z_1, z_2)$ satisfying these conditions is known as *2-D quadrature mirror filter bank*. These filters are expressed in function of the separable filter $H(z_1, z_2)$. The separability of $H(z_1, z_2)$, stated in the last equation, reduces the two-dimensional filtering problem to one-dimensional filtering, so that conventional 1-D QMF techniques can be used.

Appendix B is devoted to the analysis of the 1-D QMF filter bank. Quadrature mirror filters are designed to perform exact cancellation of aliasing in the reconstructed image $\hat{x}(m, n)$. QMF's are not perfect reconstruction filters (unless the order is one or less), but they have linear phase and the distortion can be made very small. Other proposals of filter banks for subband analysis and synthesis are also discussed in appendix B. The filters proposed by Smith and Barnwell [109], called *conjugate quadrature filters* (CQF's), have the perfect reconstruction property, but non-linear phase. Besides, the larger oscillations (ripples) of the step response of CQF filters produce more ringing effects than QMF's. Another approach, the *wavelet decomposition* [88], performs a smoother (regular) filtering of the original images. 'Wavelet' filters present high attenuation without oscillations in the stop-band, but their cut-off frequency is low thus resulting in a certain amount of blurring distortion along the edges. Finally, the family of filters called *asymmetrical filter banks* (AFB's) proposed by Egger and Li [27] are linear phase and perfect reconstruction, but they have smaller frequency selectivity and are not free of the ringing effect either.

We have chosen the set of 8-TAP QMF Johnston filters, designated as 8 A in [48], to implement the region-based subband decomposition. The characteristics of the frequency responses of these filters are given in appendix B. Johnston filters have been widely used in the context of subband image coding and have good reconstruction properties. These filters are FIR, even in length and linear phase. In addition, due to the symmetry property of the coefficients of the low-pass/high-pass filter pairs (p. 167, eq. B.1.3), the convolution operations can be implemented efficiently with the 'polyphase' structure [119], what reduces the computational load by 50 percent.

5.2.2 2-D Separable 'pyramid' subband decomposition

The filters $H_i(z_1, z_2)$ defined by eqs. 5.1–5.4 are related through the separable filter $H(z_1, z_2)$. When these relations hold, the separability property of eq. 5.5 is a necessary and sufficient condition for the cancellation of the 2-D aliased components (p. 173, eq. B.21). Furthermore, separable QMF filters are much more efficient in terms of computational load.

In order to obtain a four band decomposition as the one shown in Fig. 5.1, the basic frequency splitting step of the 2-band 1-D QMF filter bank is applied twice, along the rows and along the columns of the input image. The decomposition of the input image can be extended to more than four bands by repeating the analysis process on each sub-band in a

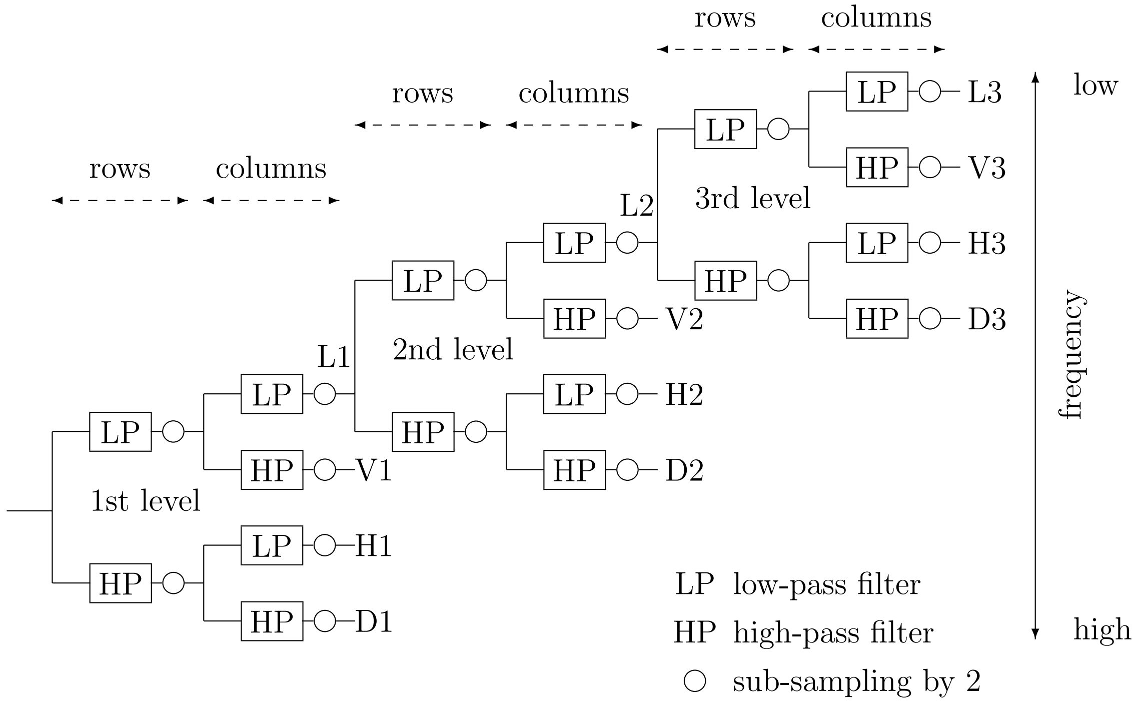


Figure 5.2: Ten-band pyramid subband decomposition

tree-structured manner. Fig. 5.2 shows an example of a 2-D separable QMF filter bank that provides a 10-band 'pyramid' subband decomposition. It is a non-uniform decomposition of the frequency spectrum where only the lowest band is further decomposed by the 2-D QMF system. At each level of the decomposition, the four bands are named with one letter within the set of L, V, H and D, according to their frequency contents:

- 'L' stands for the low-low frequency band,
- 'H', for the high-horizontal band,
- 'V', for the high-vertical band and
- 'D', for the diagonal band.

A number following the letter indicates the decomposition level where the subband has been obtained. As illustrated in the idealized frequency diagram of Fig. 5.3, such a cascaded application of the QMF filter bank partitions the frequency domain into octave-spaced oriented subbands. The higher frequency bands may be further decomposed as well but, for low bit-rate applications, a finer decomposition of the highest frequency bands leads to greater distortion effects, whereas the compression gain obtained is not significant enough to justify such distortion [35]. This can be explained because of the weak correlation observed in the high-pass filtered subbands.

5.2.3 Region-based subband analysis

Let us assume now that the original image has been divided into regions, which ideally correspond to the objects in the scene. Then, the texture inside each region may be coded independently by a subband coding scheme. The available bit-rate may be distributed inside each region over the different subbands, according to their frequency contents. The bit-rate may be distributed also among the partition information and the region contents (textures), according to their relative importance, and among the regions as well, if there exists any criterion –subjective or imposed– that marks some of them as more relevant than the others. The regions may be rather large, thus reducing the coding overhead of transmission of the partition information.

As pointed out at the beginning of this chapter, from the point of view of subband coding it is actually more interesting for the segmentation to be aimed at the location of the region boundaries, along the sharp amplitude transitions or strong edges of the image, than at the objects themselves. If the pixels at both sides of sharp transitions belong to different regions, the subband coding scheme will be able to represent efficiently the remaining less important edges and fine textures with good reconstruction quality (without significant ringing artifacts) even at low bit-rates.

The standard filtering techniques for subband analysis and synthesis, cannot be used straightforward to code the texture of arbitrarily shaped objects. The reason is that a k-level decomposition requires the regions to consist of rectangular blocks of sizes $\alpha 2^k \times \beta 2^k$, where α and β are two positive integers. In such case, k steps of 2:1 critical down-sampling can be conveniently carried out. In general, such rectangular regions will not correspond to the shape of the objects in a scene.

The result of a segmentation process is a partition represented by a *label image*, with each label corresponding to a different region. In order to perform subband analysis of a given region, the decomposition of the label image has to be defined beforehand. The standard 2-D separable subband filter bank decomposes the input image into four bands, L, V, H and D, in the frequency domain, as shown in Fig. 5.3, corresponding to four 4:1 critically sub-sampled sub-images in the spatial domain. Now, we wish to decompose the image into four bands, so that, in the spatial domain, the sub-image corresponding to each subband is subdivided into

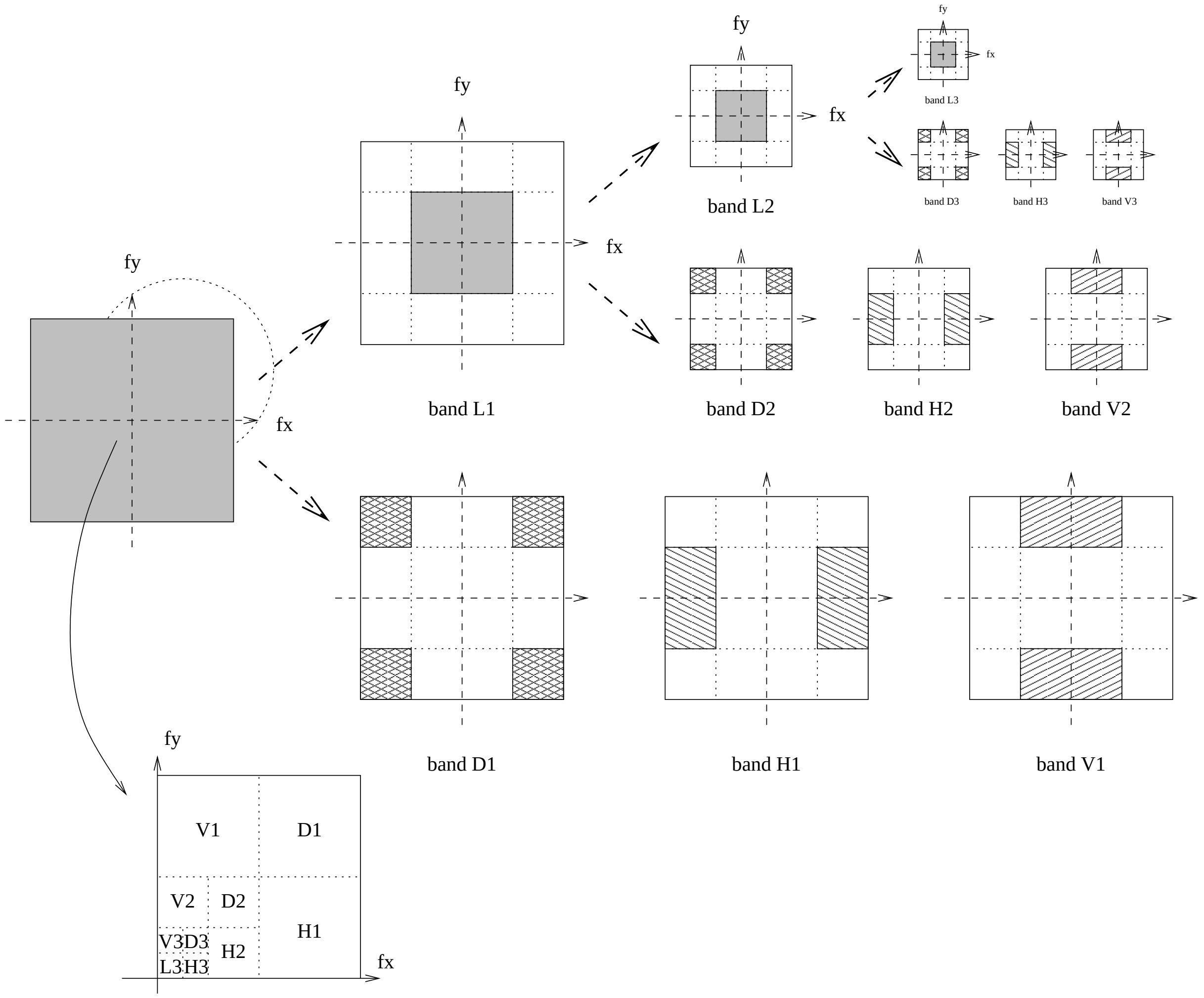


Figure 5.3: Ideal partition of the frequency domain by pyramid subband decomposition

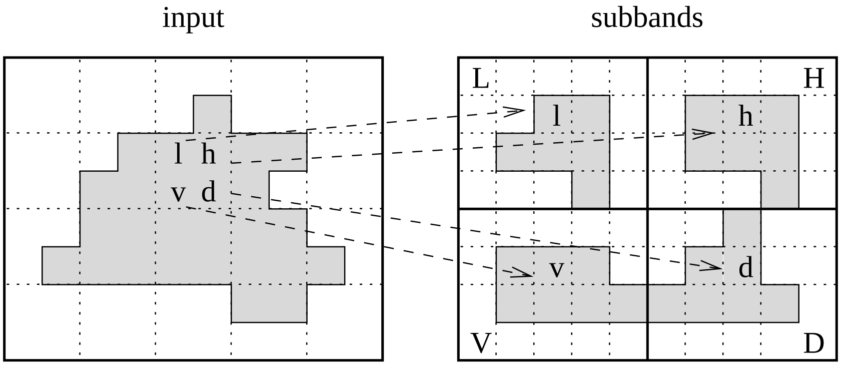


Figure 5.4: Decomposition of the label image

regions as well. Four 'child' regions of the decomposition correspond to one 'parent' region of the original image. There are two conditions that the decomposition of the label of the parent region into four child labels should fulfill:

- It must be 'invertible', so that the parent label can be losslessly reconstructed from the four children.
- The total number of pixels of the four child labels must be equal to the size of the parent label (critical sampling condition).

This can be achieved by the method depicted in Fig. 5.4. The original label image is subsampled four times by 4:1 to get the labels of the four sub-bands. The three sub-sampling schemes used for the high-frequency sub-bands are shifted in space by one pixel either right, down or both, so that for each block of 2 × 2 pixels of the parent label the upper-left pixel is put into the L band, the upper-right in the H band, and so on. Notice that for such an irregular shape (with respect to its size), the child regions are rather dissimilar, but the total number of pixels with the given label remains the same. Moreover, since decimators are space-variant operators, two regions of identical shape can be divided over the subband images into different children depending on their position in the image.

The decomposition method described above guarantees critical sampling and 'perfect reconstruction' of the parent labels. With respect to the region contents, i.e. the texture, there exist a number of different techniques for the extension of finite length signals that enables to apply *separable* subband coding on arbitrarily sized regions. However, up to the author's knowledge, only one method of signal extension allows critically sampled subband decomposition of *arbitrary* length signals while preserving the perfect reconstruction property. Making use of filter separability, it can be applied to *arbitrarily* shaped 2-D regions. This method, due to Barnard et al [4], is a signal-adaptive symmetric extension method that will be explained in the sequel.

5.2.4 Symmetric signal extension for perfect reconstruction through critical sampling [4]

Separable filter banks can be used for the filtering and down-sampling of arbitrarily shaped regions,, so that the filtering and down-sampling is performed per segment line. A *segment line* is defined to be a horizontal (or vertical, in the case of column filtering) sequence of connected pixels with the same label. Initially, a one dimensional filtering is considered, but the results hold for higher dimensions.

Extension of 1-D finite length signals for subband coding

The problem of signal extension for subband coding of images has been considered by different researchers [50], [108] in order to solve the following problem:

> Applying subband filters directly to images, by linearly convolving the rows and the columns and decimating, increases the overall number of pixels. The problem is that the linear convolution of an $N \times N$ image with an $L \times L$ filter results in a larger image of size $(N + L - 1) \times (N + L - 1)$ with an aggregate number of pixels. This is generally undesirable in a compression application because of the increase in the number of samples to be coded and transmitted. The critical subsampling principle is not fulfilled because the image is enlarged at the borders. For example, some simple calculations show that a 10-band pyramid subband decomposition by 8-TAP QMF filters leads to a total of 73.056 pixels to be coded for a 256 × 256 image when the $L - 1$ additional pixels generated at the image borders are not truncated at each level. The increase in the number of samples is about 12 percent, but with 16-TAP filter banks, the increase is 25 percent. Obviously, for smaller images –as the regions obtained in a segmentation are smaller– the increase of the number of pixels would make such coding technique completely useless. If the number of samples is enforced to be equal to that of the input image, the subband images must be truncated (windowed down to $N \times N$ samples) and, then, the information loss leads to distortion of the reconstructed signal.

To alleviate this problem, the input image signal has to be extended in an appropriate way before separable filtering, so that the information loss is minimized. Five types of signal extension can be considered for a 1-D signal of even length N [50]:

1. zero padding: the signal is assumed to be zero outside its support
2. circular extension: the signal is periodically replicated with period N
3. replication of boundary values: the signal is made continuous at the ends by repeating the first and the last sample values to infinity
4. symmetric extension: similar to circular extension, but this time the period is $2N$. This is achieved by extending the signal by its mirror image, whereby it becomes symmetric around the boundaries, and then periodically replicating the result.
5. doubly symmetric extension: the signal is made symmetric not only in space but also in amplitude by taking the border point as a symmetry axis in the amplitude-space coordinates of the representation plane.

The two first methods create a discontinuity at the signal borders, whereas methods 3–5 maintain continuity. For method 5, the first derivative is continuous at the boundary as well. It can be shown [108] that perfect reconstruction[2] is possible if, after filtering a signal of length N, the low-pass and the high-pass subband signals can be determined from a subset of $N/2$ samples. This is easily achieved with periodic extension methods 2–5 if linear phase filters are used in the analysis/synthesis system[3]. From a frequency domain perspective, the circular extension method can be modeled in terms of the product of DFT's and the symmetric extension method can be thought of in terms of a kind of $2N$-point DCT [108].

The only methods that achieve aliasing cancellation are, thus, the circular extension method (2) and the symmetric extension methods (4, 5). However, for the circular extension method, the discontinuity introduced in the borders of the signal yields artificially high amounts of energy in the high-pass band, compromising the coding gain of the subband scheme. On the other hand, the doubly symmetric extension presents a higher computational load but does not show significant improvements with respect to the symmetric one. It has been found [108] that for low-bit-rate coding applications, the symmetric extension method performs the best.

[2] In the sense of alias cancellation and optimization of the overall response of the system, as carried out in the design of QMF filter banks

[3] Given that, then, the filtered and down-sampled signals are also periodic and/or symmetric.
Some well known properties of signal theory apply here. First, the impulse response of linear phase filters presents symmetry characteristics. It must be either symmetric (even symmetry) or anti-symmetric (odd symmetry)[63, p. 198]. Second, the response of a linear system to a periodic input signal, is a periodic signal as well with the same period [63, p. 309]. And, third, if the input signal is a symmetric function and the analysis filter presents a symmetric impulse response, then the filtered signal is also symmetric. If the filter is anti-symmetric, then the output is necessarily anti-symmetric[63, p. 43].

Nevertheless, when the signal length is not a multiple of 2^k and the number of samples in the decomposition is required to be kept constant (critical sampling), none of these methods preserve the perfect reconstruction property. The perfect reconstruction may be achieved by the adaptive symmetric extension proposed by Barnard et al in [4]. Their signal-dependent symmetric extension method makes possible the subband decomposition of 1-D signals of any length N up to any level k and, in principle, up to N subbands of length 1, without losing the perfect reconstruction property. Of course, due to the arbitrary lengths of the segment lines obtained from the region labels, this method fits the needs of region-based subband coding.

Application of symmetric signal extension to 'segment lines'

Although the symmetric periodic extension technique developed by Barnard et al [4] depends on both the signal values and the properties of the impulse response of the filters, only its implementation for a particular filter bank of separable symmetric filters will be shown here. Let us assume that the QMF filter bank presents the symmetry properties expressed in eq. B.1.3 of p. 167. Thus, the low-pass filter is of even length and linear-phase and its impulse response is symmetric. The corresponding high-pass filter is anti-symmetric.

The segment lines can be divided into four classes, depending on the possible combinations of the parity of the column (or row) number at which the segment line starts and ends. There are two possibilities for the start- and end-points of a segment lines, namely *even start/odd start* and *odd end/even end*. The even-start odd-end case is illustrated in Fig. 5.5.

The top rows show the original pixel values, which are denoted by lower case letters. Outside the signal support (indicated by the lines below the values) the signal is symmetrically extended by mirroring, which is visualized by using the same lower case letters when the same signal values appear. The symmetry axis lies halfway between two samples. This results in a smooth extension so that no sharp transitions are introduced.

The rows in the middle represent the filter positions for the convolution operation. The filter coefficients are written in upper case letters and symmetry is symbolized in the way the letters are repeated. For each filter position, the inner product is computed with the filter and the part of the extended signal in the top row, directly above the filter. The values resulting from the inner products are put in the central row. For example, k is the result of the inner product of the low-pass filter in its first position with the part of the signal denoted by $c, b, \ldots, e$. The down-sampling is illustrated by the 'asterisks' of the central row. The values outside the support of the down-sampled signal show which results would have been obtained if the convolution would have been computed over more positions than those of the down-sampled signal support. Since the filters are symmetric, it can be observed that

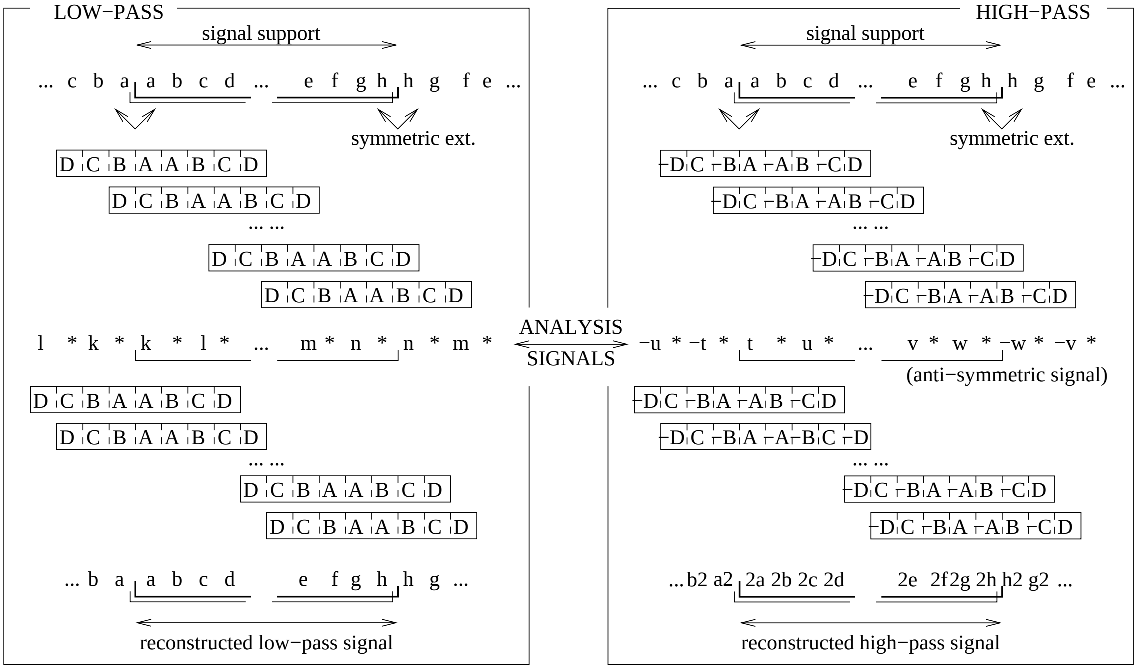

LOW−PASS
signal support
... c b a a b c d ... e f g h h g f e ...
symmetric ext.
D C B A A B C D
... ...
l * k * k * l * ... m * n * n * m *
ANALYSIS
SIGNALS
... b a a b c d e f g h h g ...
reconstructed low−pass signal
HIGH−PASS
signal support
−D C −B A −A B −C D
−u * −t * t * u * ... v * w * −w * −v *
(anti−symmetric signal)
...b2 a2 2a 2b 2c 2d 2e 2f2g 2h h2 g2 ...
reconstructed high−pass signal

Figure 5.5: Filtering a segment line with even start and odd end pixels

no new values appear. The symmetry of the analysis signals is even for the (symmetric) low-pass filter and odd for the (anti-symmetric) high-pass filter. Thus, perfect reconstruction is possible because all the information in the subbands is contained in the samples of the down-sampled signal support.

In the synthesis stage, the subband signals are extended as shown in the central rows and up-sampled with zeros. Note that the filters are non-causal. With the proper choice of filter delays, the total delay of the system is made zero.

Even end case Now consider that the input segment line for the current label ends at an even position. For the explanation of this case, we will refer to Fig. 5.5 as well. According to the label-splitting method depicted in Fig. 5.4, if sample h does belong to the neighboring label then the rightmost sample w of the high-pass signal will be occupied by a certain value resulting from the analysis of such neighboring label. The extension method, in this case, consists of adding one specific sample h and, then, the signal is extended as explained above. The value of the sample h in the extension will be chosen so that the subband sample w (which is not coded) becomes a fixed value. This fixed value is signal-dependent and, therefore, contains no information about the signal. As both the receiver and the transmitter know this fixed value beforehand, it does not need to be transmitted and so, the number of samples in the subbands equals the length of the original signal. In other words, critical sampling is fulfilled.

The fixed value of the sample represented by w in the high-pass subband of Fig. 5.5 is a result of the high-pass filter in the last four pixel positions, and forms a linear equation with one unknown variable h. The value for w can be fixed to zero due to the zero mean of the high-pass filter, so that, h is uniquely determined.

$$w = (-A + B)h + (A - C)g + (-B + D)f + Ce - Dd = 0 \tag{5.6}$$

then,

$$h = \frac{(A - C)g + (-B + D)f + Ce - Dd}{A - B} \tag{5.7}$$

In the synthesis stage, a zero value will be used instead of w and symmetric extension will be performed as explained above.

Odd start case Next, consider that the input segment line for the current label starts at an odd position. Barnard's solution and the method presented here differ slightly in this case. While they perform a shift to the right of the input signal in order to consider the segment line as an *even start* case, we apply the same strategy than in the previous case, but in a somewhat different manner. The advantage of this modification will be explained below.

Referring to Fig. 5.5, if sample a does not belong to the current label, then, according to the label-splitting method, the leftmost sample k of the low-pass signal will be occupied by a certain value resulting from the analysis of the neighboring label. The extension method in this case, consists of adding one specific sample a and then extending the signal as before. The value of sample a in the extension will be chosen so that the subband sample k (not coded) becomes a fixed value known beforehand.

The fixed value of sample k in the low-pass subband is a result of the low-pass filter in the first four pixel positions. Let us assume a value $k = x$ for this sample. The sample a needed for the extension can be computed as follows:

$$k = (A + C)b + (B + D)c + (A + B)a + Cd + De = x \tag{5.8}$$

thus,

$$a = \frac{x - (A + C)b - (B + D)c - Cd - De}{A + B} \tag{5.9}$$

The value for a can be fixed to any value according to the statistics of a low-pass subband signal. If this signal is assumed to be smooth, the value for k may be taken to be equal to its neighbor pixel l, then

$$a = \frac{(-A + B - C)b + (A - B - D)c + (A - C)d + (B - D)e + Cf + Dg}{(A + B) - (C + D)} \tag{5.10}$$

The disadvantage of this value for sample a is that 6 samples are needed, what increases the computational load. In addition, it will be difficult to define this value for segment lines of length shorter than 6. Given that samples f and g are at a rather large distance from sample a, instead of their actual signal values, they may be taken to be $f = e$ and $g = d$. With this, a kind of second symmetric extension is performed inside the signal itself, and a becomes

$$a = \frac{(-A + B - C)b + (A - B - D)c + (A - C + D)d + (B - C + D)e}{(A + B) - (C + D)} \tag{5.11}$$

Moreover, taking into account the values of the filter coefficients, the last term in the numerator of eq. 5.11 has only one percent of the weight of the other three terms. Therefore, sample e may be obviated and only three samples b, c and d are needed. Of course, this computational advantage will be at the expense of a certain loss, though small compared to the quantization loss. If perfect reconstruction is required, a may always be computed through eq. 5.9. In the synthesis stage, the value of l will be used instead of k before symmetric extension.

Special cases Some special cases arise when the segment line is of length $N = 2$ or $N = 1$. A detailed study of these two cases yields the following conclusions:

- For signal length $N = 2$, after applying the extension method, the value of the low-pass sample results in the average of both samples and the high pass sample is half the difference of their values.
- For signal length $N = 1$ and even start of the segment line, the signal cannot be further decomposed. The extension and filtering yields its value in the low-pass band and nothing (the not transmitted zero sample w) in the high pass. Thus, the sample is simply copied to the low-pass band.
- For signal length $N = 1$ and odd start of the segment line, the label splitting method places the label of the sample in the high-pass band. This sample is called a *single.* After the extension, filtering and down-sampling the single will result in a zero value, because of the zero mean of the high-pass filter, and, therefore, its information will be lost.

A solution to the problem of single samples is to keep the value and copy it directly into the high-pass band. Such sample contains low-pass information, and will be quantized separately. Quantization and bit-allocation of the remaining high-pass samples will be performed without the singles. This guarantees that the singles do not disturb the statistics of the high-pass signals.

Advantages The advantages of the variant of Barnard's method presented, besides the possible computational efficiency for odd start cases, are the following:

- the fact of not shifting the segment line preserves the continuity of low-pass features in the second spatial dimension
- the value of k assumed for the low-pass band constrains the values of l, m and n –which also depend on the value of a– to be smooth, so that further decomposition of the low-pass band is worthwhile, even near the region boundaries.

Similar symmetric extensions may be computed for 2-D signals for non-separable filters, at the cost of extra computational complexity. The extra samples depend then on each other, and their computation requires, in that case, solving a linear system of as many dimensions as the filter length.

5.3 Bit allocation and perceptual quantization

The main ability of subband coding is the compression of visual information by allocating different numbers of bits to the quantizers applied to each subband based on perceptual criteria. In principle, bit allocation is a rate-constrained optimization problem of, given a total bit-rate, optimally assigning such quantizers to the subbands (sources) to be coded. The optimality of the solution is taken as an overall distortion measure of the reconstruction error. The only assumption that optimal bit allocation algorithms [105], [124], [84], [87] make about the rate and distortion measures R and D is to be *additive*. That is, they can be written as separate sums of, respectively, the individual rates r_n and distortions d_n of the subbands $n = 1, \ldots, N$:

$$D = \sum_{n=1}^{N} d_n \qquad R = \sum_{n=1}^{N} r_n \tag{5.12}$$

The distortion d_n may be the result of any arbitrary distortion measure (not only the mean squared error as it is normally used). In addition, no restrictions are imposed on the possible choice of quantizers. They can be of any type, and even vector quantizers.

The squared error is an additive measure over the subbands, since the filtering and upsampling performed at the synthesis stage are linear operations. That is, the total squared error distortion can be computed either on the reconstructed region or as the sum of the individual squared error distortions introduced by the quantizers in each subband, assuming normalized linear filters[4]. This situation is called a case of *independent quantization* by Ramchandran et al in [83]. They have shown that at optimality (for normalized filters) each subband should present the same ratio between distortion and rate[5].

The result of the optimization algorithm for a given bit-rate consists of a set of N quantizers, one for each subband. In order to find the best set of quantizers, one could calculate the R and D values for all possible combinations of M quantizers over the N bands to obtain M^N different (R, D) pairs. By analogy to rate-distortion theory, where $R(D)$ curves are known to be convex, if we represent the (R, D) pairs by their $R(D)$ plot, the optimal bit allocations can be defined as those points that lie on the lower convex hull of all possible bit allocations [124]. For the particular case of subband coding, the number of possible bit allocations is rather large. Typical values for the number of bands and quantizers could be $N = 10$ and $M = 10$, and it is not reasonable to compute 10^{10} combinations to find the convex hull.

[4] Normalized linear filters filters which preserve energy. If the filters are not normalized, but only linear, the individual distortion terms should be weighted accordingly to the filter gains.

[5] Actually, they have investigated the case of an open-loop Laplacian pyramid scheme, which is a particular case of subband coding without down-sampling of the high frequency bands (not preserving the critical sampling condition).

The bit allocation algorithms referred above are designed to solve this problem but do not require the calculation of all possible bit allocations. Only the $N \times M = 100$ individual measures of r_n and d_n for each subband are necessary. An interesting fast implementation is the one proposed by Westerink et al in [124]. It allows the search of the optimal quantizer set for a given rate, starting from the lowest distortion combination and progressing in the direction of increasing distortion or, vice-versa, from the smallest rate and in the direction of increasing distortion.

5.3.1 Bit allocation for region-based subband coding

If subband analysis has been performed over a given segmentation mask, then the bit allocation problem can be formulated over smaller sources (which correspond to the objects in the image) whose rate and distortion measures are additive as well. As discussed before, each subband will be divided into regions according to the label splitting method of section 5.2.3. Thus, *the quantizers may be chosen for each region of each subband.* The capability of adapting the quantizers to the spatial contents of the image as well as to the frequency content is what makes region-based subband coding attractive.

On the one hand, this possibility represents an increase in the complexity of the bit allocation algorithm. Depending on how many regions have been defined in the label image, the computation of the rate and distortion measures $r_n(i)$ and $d_n(i)$ for all bands n, regions $i = 1, \ldots, S$ and associated quantizers may be rather large. Despite the large number of values $d_n(i)$, they can be efficiently computed because each one will be over a small number of pixels (assuming that the measure is additive over the pixels as well). However, the rate values $r_n(i)$ must be computed at the output of the quantization buffers either by directly applying an entropy coder or by a close estimation of their performance from the statistics of the quantized images. Since such measure is clearly non-additive over the samples, it should be computed independently for each case.

On the other hand, the choice of quantizers for each region and subband must be transmitted to the decoder. This may represent a significant amount of overhead information, as it is the case with any coding system with a high degree of flexibility. For example, in the case of 10-bands and 10 possible quantizers, about 30 bits per region are necessary to specify the optimal quantizer set, independently of the size of the region. Only the overhead information regarding the quantization choice (the bulk of the overhead information is assumed to be the contour information) would amount to 0,1 bpp for a small region of 300 pixels. Therefore, the set of possible quantizers has to be restricted in some way for very low bit-rate coding applications.

Interaction with segmentation

The interaction of the bit-allocation strategy with the segmentation result is an important issue that must be pointed out before the definition of a suitable strategy for bit-allocation in a region-based subband coding scheme. From the rate-distortion point of view, the gain obtained in some cases through the separate coding of two regions –by adapting the filters with the symmetric extension technique and the quantizer selected for each region– may not be worthwhile. For a given distortion value, the savings in bit-rate could be smaller than the increase in the overhead data due to contour information and quantizer choices. In such case, it may be better the merging of the two regions than the independent subband analysis and corresponding coding of each one. A similar reasoning can be used for the opposite case of splitting two regions.

Only when the optimization algorithm for the bit allocation is applied, the parameters needed in order to take this kind of decisions are available. However, any change in the label image, would imply to repeat the subband analysis process, the quantization, the rate-distortion computations and, finally, to re-apply the bit allocation algorithm. The interaction of subband coding with segmentation is managed by a *higher level* rate-distortion optimization algorithm [87], [69], [75]. Subband coding can be applied and optimized for each region independently by means of a *lower level* rate-constrained optimization algorithm.

Two-level rate-distortion optimization This algorithm is explained in in more detail in appendix C. The higher level algorithm works with several hierarchical segmentation proposals (called the *partition tree*). The candidate regions of these proposals are coded with the texture coding technique (here, subband coding) at different target rates. From the rate-distortion figures resulting from the coding of each candidate region, and for a given overall target rate, the decision of the higher level algorithm results in a specific segmentation of the input image and an optimal set of rate-distortion pairs for each region.

Therefore, the bit-allocation strategy for subband coding may be confined to each individual region with a given set of rate constraints. If the regions are meant to represent different objects in the scene, they may be coded independently. The quantizer choice for the subbands of one region do not influence the coding of the neighboring regions, given that their spatial domains are disjoint.

5.3.2 Perceptual considerations for bit allocation

As pointed out in the introduction of the current section, the optimal solution of the bit allocation problem results in a distribution of the available bit-rate over the subbands such that each band presents the same distortion. This forces the choice of quantizers to be adapted to the frequency contents of the region. However, it has been said that the bit allocation should be made adaptive to the frequency contents of the regions *based on perceptual criteria.* This condition is not likely to be reached if the only possibility consists of having the same distortion for all the frequency bands.

The distortion measure $d_n(i)$ for subband n of region i may be any measure of distortion, provided that it is additive over the subbands of that region. Therefore, one possible solution to obtain a 'perceptually-motivated' bit allocation over the subbands is to use a 'perceptual' measure of distortion. Such measure should be, at least, frequency-dependent, i.e. based on the frequency response of the human visual system, so that different weights should be given to the errors introduced by the quantizers in the different subbands. By using these weights, the subband system may be designed so that the noise remains below the *just noticeable level* (JND) of perception because of its spatial frequency distribution. For very low bit-rate applications, where a certain amount of distortion must be tolerated, the noise should be distributed so that it should be minimal and equally 'visible' at all frequencies. This is known as a *minimally noticeable distortion* (MND) profile [46]. Subband coding schemes provide a natural framework for rate-constrained allocation with 'unequal' distortion values on the subbands if perceptually-weighted distortion measures are used.

The frequency sensitivity of the human visual system, described by its *modulation transfer function* (MTF), has been employed in different works in order to obtain perceptually-weighted measures of distortion for subband coding [92], [80], [20]. The *homomorphic model* proposed in the introduction of this report (section1.2.1, p. 9) that transforms the image in a *perceptually flat* domain where the unweighted error measure (MSE) is useful, can be achieved by tuning the gains of the subband analysis filters according to the MTF before quantization and using reciprocal factors in the synthesis stage.

As pointed out in chapter 1, the main problem of the previous approach is that the MTF is a *global* property of the images and does not consider the masking properties related to the *local* scene content. In order to achieve the highest subjective quality at a given bit-rate, these properties of our visual system must be utilized. Directional edge decompositions [54] and 'geometrical' vector quantizers proposed for subband coding [81] are examples of coding systems that follow this second approach of considering local properties of the scene.

Object-oriented and edge-oriented coding systems exploit the masking effect of sharp transitions based on a certain model of the image. Region-based subband coding can be considered among these ones. The fact of considering the information about the location of sharp transitions in the label image can be made available to the quantizers in order to perform spatially adaptive quantization. A similar approach has been reported in [55], where edge features appearing in the subband images are adaptively quantized.

Some constraints must be imposed to the bit allocation algorithm in order to introduce perceptual criteria in the coding of subband regions. These constraints may be classified as frequency (global) constraints and spatial (local) constraints.

Frequency constraints

Perhaps the most interesting result of the subjective measures of the MTF is that the sensitivity of the eye for frequency gratings oriented at a diagonal direction is about 3 dB less than for vertically and horizontally oriented gratings [80]. In order to get a perceptually consistent distribution of the distortion among frequency subbands, the possibilities to be tested by the optimization algorithm are restricted to quantizer sets having coarse quantization parameters for the highest frequency subbands and even coarser for the diagonal subbands.

Spatial constraints

The proximity to region boundaries must be considered as well. This can be done by computing the distortion of the subbands for all pixels of the label except for those located at a certain distance of the region transitions. Quantization errors produced at such pixels – usually larger than in other locations due to the proximity of the sharp transition– are not considered in the distortion measure[6]. Such perceptual distortion computation is performed for regions having a certain minimal size compared to the number of border pixels[7].

5.3.3 Design of perceptual quantizers

Woods [128] proposed the use of DPCM to encode the image subbands. Based on auto-covariance measurements of each subband, other authors have shown that only the low-pass subband presents large coefficients in its auto-covariance matrix [36], [125] in a one level

[6] The considered pixels are those whose neighbors belong to a different region (border pixels). The distortion measure is computed 'per pixel'. Therefore, from the point of view of the distortion measure, the border pixels are assumed to have the same averaged quantization error than the rest.

[7] So that the measure of distortion measured from non-border pixels is statistically significant.

subband decomposition. Therefore, only this band could really benefit from DPCM encoding, while the other bands may be encoded using PCM. For higher level subband decompositions, the prediction gain of DPCM over PCM is smaller [35]. In addition, the portion of pixels in the low-pass subband may be actually very small (about 1.5 percent in a 10-band decomposition). In that case, DPCM coding of the low pass subband is not worthwhile.

An optimal approach for PCM coding, in the minimum mean squared error sense, is to design a quantizer matching the probability density function of the input image (close to Laplacian in the case of the high-pass subbands). However, it has been observed [36] that such a quantizer is not suitable in the perceptual sense. This observation leads to the design of quantizers for the subband coding with the following characteristics (see Fig. 5.6):

- a center dead zone $\pm d$ where the input values are mapped to zero to eliminate picture noise
- a limited quantization range between two thresholds $\pm t$ to cover moderate contrast changes
- uniform quantization with L levels within the active range
- two saturation values $\pm y$ for pixel values above or below the thresholds $\pm t$.

The quantizer is the part of the overall system where the perceptual coding may be explicitly implemented. The quantization function defines the levels of just-noticeable distortion and multiples of those, so that the visibility of the introduced distortion is minimal. A bank of quantizers parameterized through the values of $\pm d$, $\pm t$ and L is used by the bit allocation algorithm in order to perform optimization of the quantizer set to be applied to the subbands. This leads to a solution that may not be optimal, in the minimum mean squared error sense, but shows less noticeable artifacts.

Quantizers for color signals

The encoding of color images directly in the RGB domain is not very efficient. It is necessary to choose a proper color domain and a corresponding error criterion. A compact representation of color images in energy terms is achieved by transformations such as the PAL television signals YUV or the NTSC signals YIQ [98]. These representations are known to match the human visual perception properties of hue and saturation [82]. Their suitability for subband coding has been investigated by Westerink et al in [125]. The mean squared error measure they propose for the YUV color domain is the following:

$$D_{YUV} = E\left[(\hat{Y} - Y)\right] + 0.3E\left[(\hat{U} - U)\right] + 0.3E\left[(\hat{V} - V)\right] \tag{5.13}$$

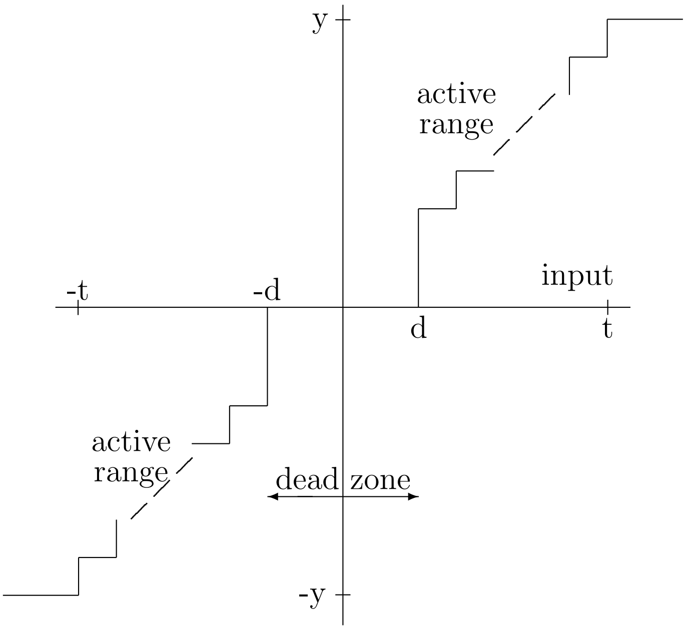


Figure 5.6: Parameters of the employed quantizers

The weights applied to the mean squared error of the color components have been optimized from test experiments. These weights indicate the subjective trade-off between color errors and luminance errors. Taking the measure D_{YUV} for the bit allocation algorithm leads to a proportional division of bits between the chrominance and the luminance components according to these weights.

Furthermore, the spatial frequency response of the visual system to color signals shows that the cut-off frequency for the chrominance components is smaller than for the luminance component [82]. This fact has been used to justify a drastic reduction of the high frequency information for the UV components. The CCIR recommendation 601 and the CCITT common image format (CIF) for example, define digital video formats where the luminance and chrominance signals are sampled at the ratio 4:2:2. Actually, the energy of the chrominance high frequency bands after subband analysis is very low and, for high compression applications, the optimization algorithm hardly allocates any bit to these bands. Therefore the chrominance high frequency subbands are simply not encoded [35], [125].

Buffering of quantizer outputs

The output of subband quantizers must be entropy coded to benefit from the highly picked distribution of quantization values. Variable word length entropy coders such as the Huffman coder [44] and run-length coding [130] have been frequently used in subband coding schemes. In [57] the advantages of using arithmetic coding [126] for the high frequency subbands, due to the contour-like appearance of the features present in such bands, have been discussed. We propose the use of arithmetic coding for the buffers resulting from the quantized subbands as well. However, the pixels belonging to each region in a given subband must be grouped together before coding, to let the coder extract the redundant information due to the homogeneity of the texture content of such region. The subband pixels must be scanned in order to put them in the buffer before entropy coding. The scanning procedure is illustrated in Fig. 5.7. The idea for the vertical scanning of the horizontal band has been taken from Gharavi [35]. Such technique follows the vertical edge features appearing in the horizontal band so that the entropy coder yields a better performance. We did not find any improvement in diagonally scanning the diagonal band, as suggested by Gharavi. A possible explanation could be the presence of both diagonal directions in the edge features appearing in this subband.

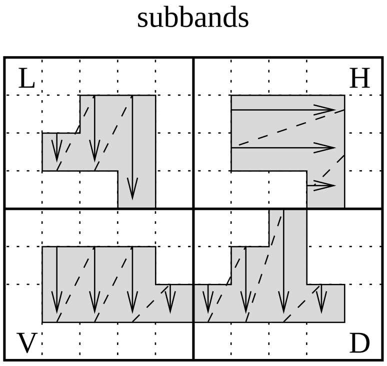


Figure 5.7: Procedure for scanning region-based subband pixels

5.4 Examples of region–based subband coding

This section is devoted to some examples of application of the region-based subband coding technique. The original image that has been chosen contains both sharp transitions and homogeneous regions with fine textures or smooth areas, so that the main questions discussed so far can be clearly illustrated.

5.4.1 Subband coding of a single region

In the first example, one single region has been extracted from a segmentation of the *cameraman* image. The original content of the region and its label are shown in Fig. 5.8. Notice that the content of this region is not very homogeneous. Besides the high frequency texture of the grass, some sharp contours appear within the region, as well as some objects in the background with a little blurring.

Subband analysis is performed with a 10-band (3-level) pyramid subband decomposition. Separable 2-D QMF Johnston filters are applied inside the region. When the convolution window crosses the region borders, the symmetric extension technique for segment lines explained in section 5.2.4 (p. 96) is used. The label splitting method of subsection 5.2.3 is applied to the image label in parallel, so that the pixels resulting for each subband are labeled as well. The results of the analysis are presented in Fig. 5.9.

Notice that some pixels can be disconnected from the main label in the subbands. This is due to the down-sampling procedure and cannot be avoided. If these pixels become 'singles' in the high-pass band or isolated pixels in the low-pass band, they will not be further processed

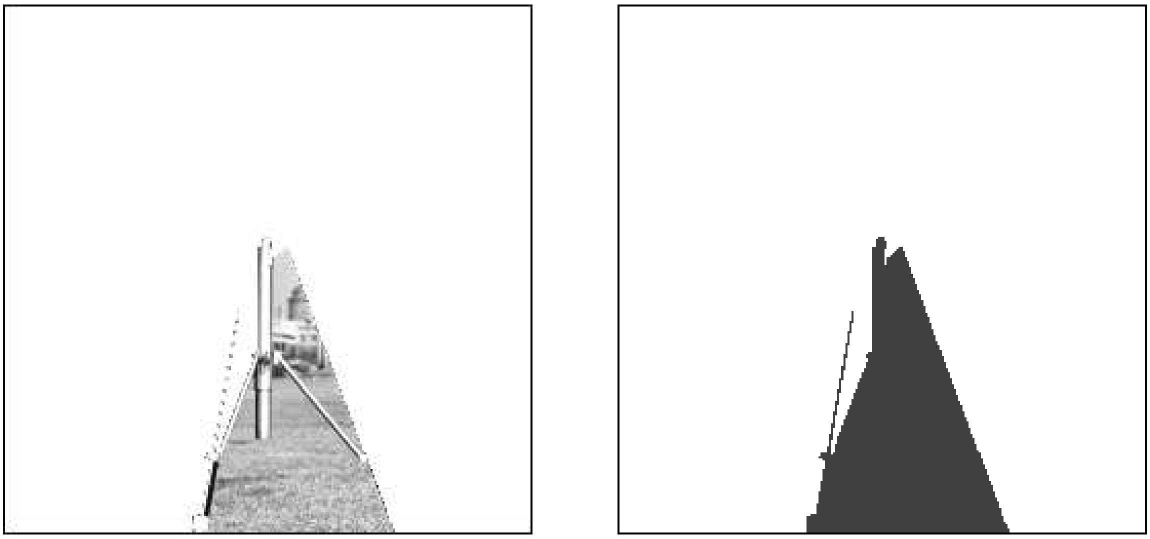


Figure 5.8: Original texture of a single region and its label

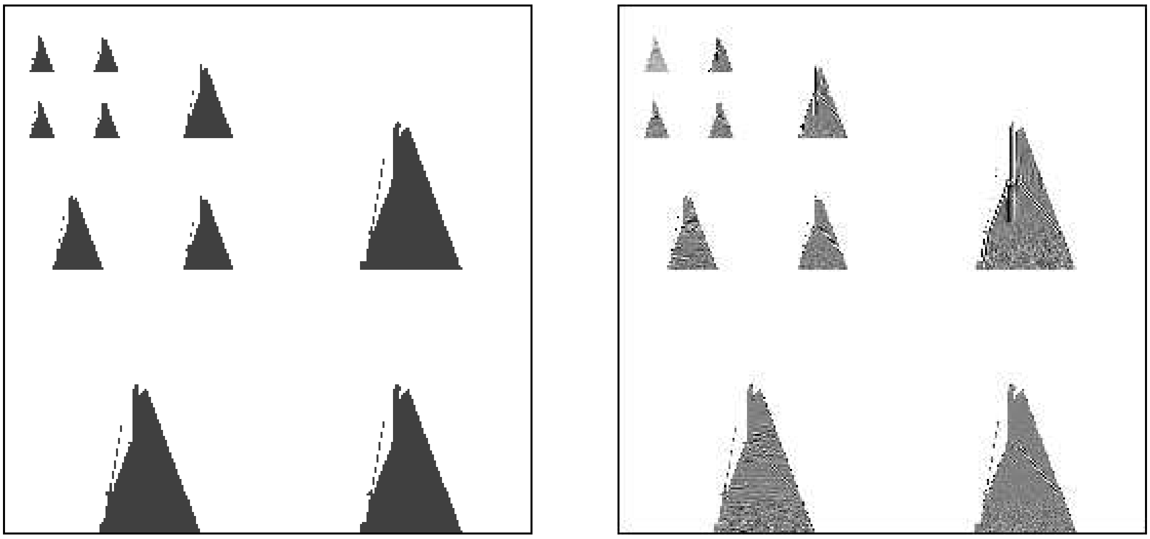


Figure 5.9: Analysis labels for a single region and the corresponding subband images

by the filters. The high-pass subband images have been stretched from ±32 to ±128 and shifted by 128 so that their contents can be more clearly shown. It can be seen that the sharp transitions contained in the interior of the region are spread over several frequency bands. On the other hand, the pixels located at the steep slopes of the edges placed at the borders of the region[8] also yield significant amplitude values in the high frequency subbands.

The decomposition of Fig. 5.9 is almost lossless, because, as indicated in appendix B, Johnston QMF filters are optimized to be nearly a perfect reconstruction filter bank. The synthesis from the non-quantized subbands yields a reconstruction mean squared error of 0.29 for the given example, partly due to the nearly perfect reconstruction property and partly, to numerical errors of the finite precision arithmetic operations.

The next step of the subband coding scheme is quantization. The quantized subbands are shown in Fig. 5.10 for two optimized sets of quantizers. The quantized subband images are displayed using the representation values for each quantizer level, stretched and shifted as explained for Fig. 5.9. The optimization has been performed according to the perceptual bit allocation technique described in section 5.3 for two target rates of 0.8 and 0.4 bits per pixel. The acronym 'bppr' indicates bits per pixel of the region. The imposed rate constraints are rather low for the texture information of such a complex region. Notice that most of the high frequency pixels have been set to zero by the quantizers.

The quantized subbands of each decomposition level are independently scanned and written into four different buffers. According to the subband names given in Figs. 5.2 and 5.3, these buffers contain, respectively,

- the first level high-frequency bands V1, D1 and H1,
- the second level high-frequency bands V2, D2 and H2,
- the third level high-frequency bands V3, D3 and H3,
- and the lowest frequency band L3.

The horizontal bands, H1, H2 and H3, are scanned in the vertical direction to benefit from the redundancy of the vertical structures appearing in these bands. The buffers are entropy coded with a first order arithmetic coder. The contributions of each subband level to the final bit-rates, in bppr, for the two cases of Fig. 5.10, are given in table 5.1. It has been proved empirically that further decomposition of any of the resulting subbands does not yield any reduction of the given rates with respect to the current decomposition.

[8] The contours separating two regions in the label image have zero pixels width. Therefore, the edge pixels of a sharp transition between two regions are distributed between the two labels. This generates border pixels, either belonging to one region or to the other, whose statistics differ from those of the homogeneous interior of the region.

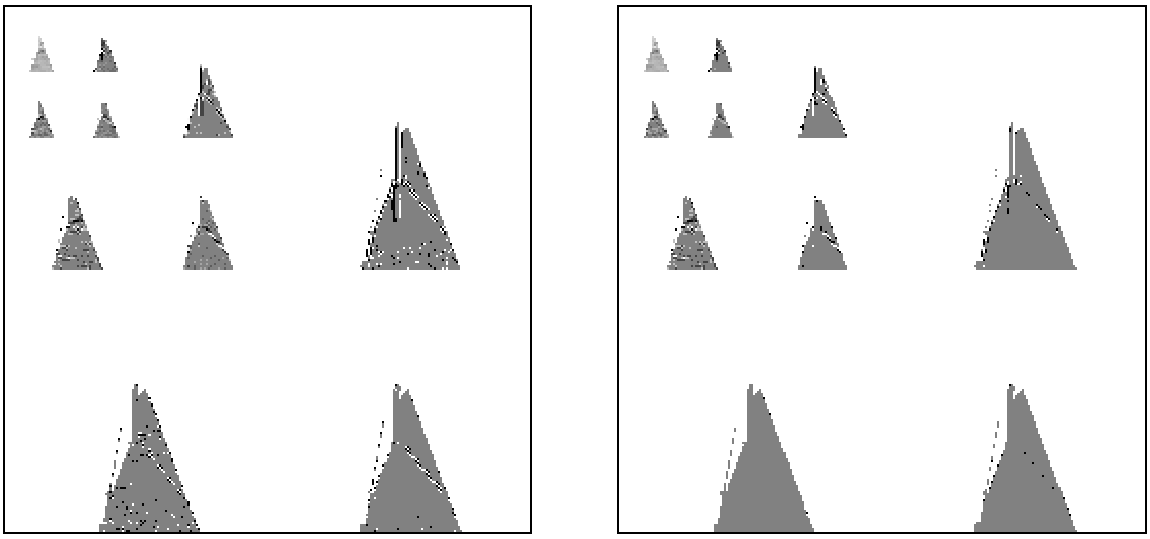


Figure 5.10: Quantized subbands for target rates of 0.8 and 0.4 bits per pixel of the region

Table 5.1: Subband rates in bits per pixel (bppr) of the region

TARGET RATE	0.8 bppr	0.4 bppr
1st buffer: bands V1, D1, H1	0.37 bppr (46%)	0.10 bppr (26%)
2nd buffer: bands V2, D2, H2	0.22 bppr (27%)	0.16 bppr (42%)
3rd buffer: bands V3, D3, H3	0.13 bppr (16%)	0.07 bppr (17%)
4th buffer: lowest band L3	0.09 bppr (11%)	0.05 bppr (13%)
PSNR	25.7 dB	23.1 dB

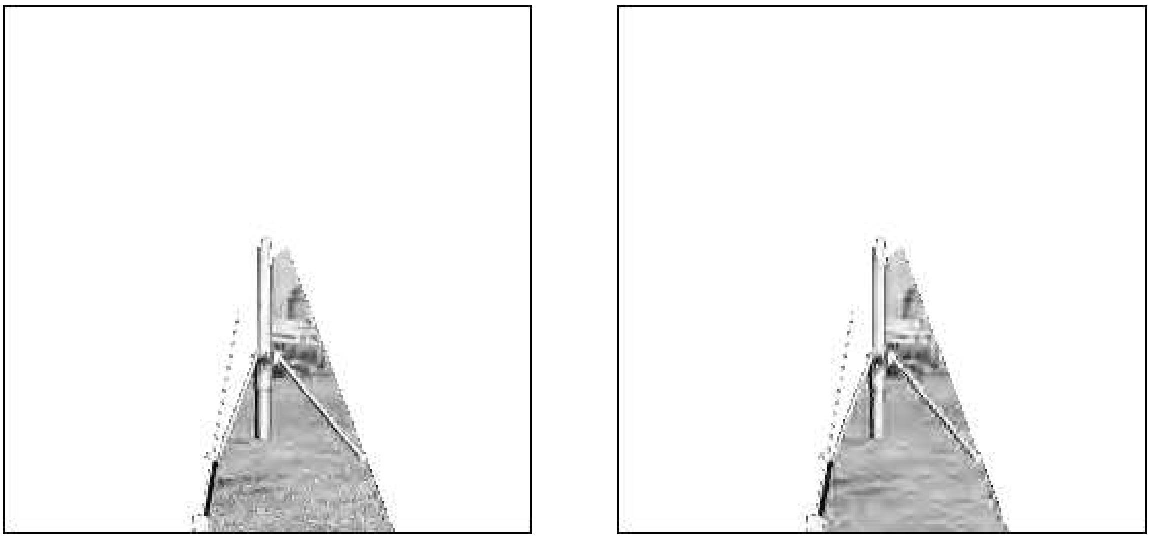


Figure 5.11: Reconstructed images for target rates 0.8 bppr and 0.4 bppr

At the decoder side, the region content are synthesized from the information contained in the subband buffers. After entropy decoding and inverse quantization, the synthesis filter bank is applied to the subband images. The label splitting is reversed in order to reconstruct the original region and, when needed, pixel values a and w fixed by the symmetric extension technique, are used as explained in section 5.2.4.

The reconstructed region contents are shown in Fig. 5.11 and the PSNR ratios for the two images, in table 5.1. At the rate of 0.4 bits per pixel, the ringing artifacts are clearly visible in the neighborhood of the strong transitions in the interior of the region, as happens in any conventional subband coding scheme working at low bit-rates. It is interesting to observe the rendering of the fine textures in both images. At the lowest rate, most of the fine texture information has been filtered out by the coding system, assuming that such information is less important from a perceptual point of view.

5.4.2 Subband coding of a segmented image

Let us proceed now to the coding of the whole segmented image. The original image and the segmentation labels are shown in Fig. 5.12. The segmentation has been performed with the morphological segmentation technique described in [94]. The label image is coded by means of an efficient chain code technique [65] that yields a total amount of 5072 bits (about 0.08

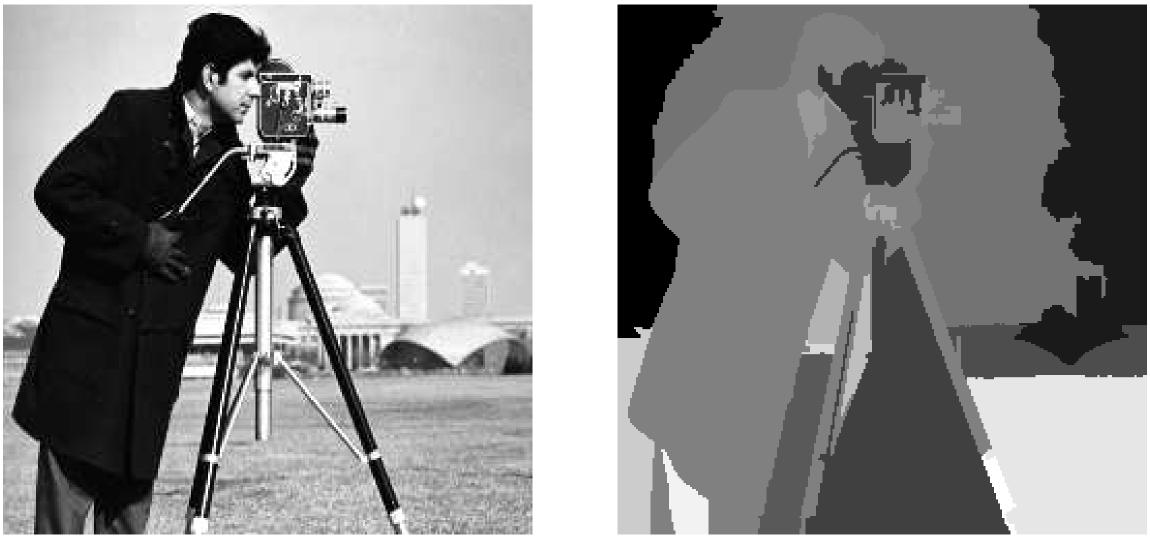


Figure 5.12: Original image and segmentation labels

bpp) for the contour information.

Subband analysis is performed for all the 18 regions of the label image. The result of the label splitting procedure is shown in Fig. 5.13 and the subband images, in Fig. 5.14. For a comparison of the performances of region-based subband analysis and conventional subband analysis (without regions), the analyzed image without considering the label information is also shown in Fig. 5.14 (right image). Please notice that the transition information has been reduced in the subbands for the region-based analysis case. This can be observed, for example, in bands H2 and H3.

Both analysis images shown in Fig. 5.14 are quantized. Then, the bit allocation algorithm is run. The rate-constrained optimization gives an optimal quantizer set consisting only of 10 quantizers for the 10 subbands in the conventional case. In the region-based analysis case, such optimization is performed *independently for each region* by the lower level algorithm described in sub-section 5.3.1 (bit allocation among frequency bands). The higher level optimization algorithm decides the rate constraints to be applied for the independent coding of each region. In this way, an optimal distribution of bits among the regions is achieved (spatial bit allocation). To summarize, the optimal quantizer sets for the subbands are found on the whole image basis in the conventional case, whereas in the region-based case they are defined on a region basis. This is the key for the adaptive capability of region-based subband coding.

The overall target rate has been fixed to 0.5 bits per pixel in both cases. The reconstructed

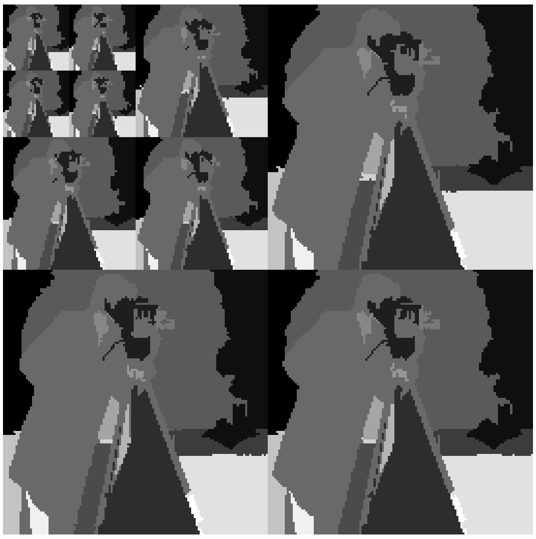


Figure 5.13: Subband decomposition of the label image

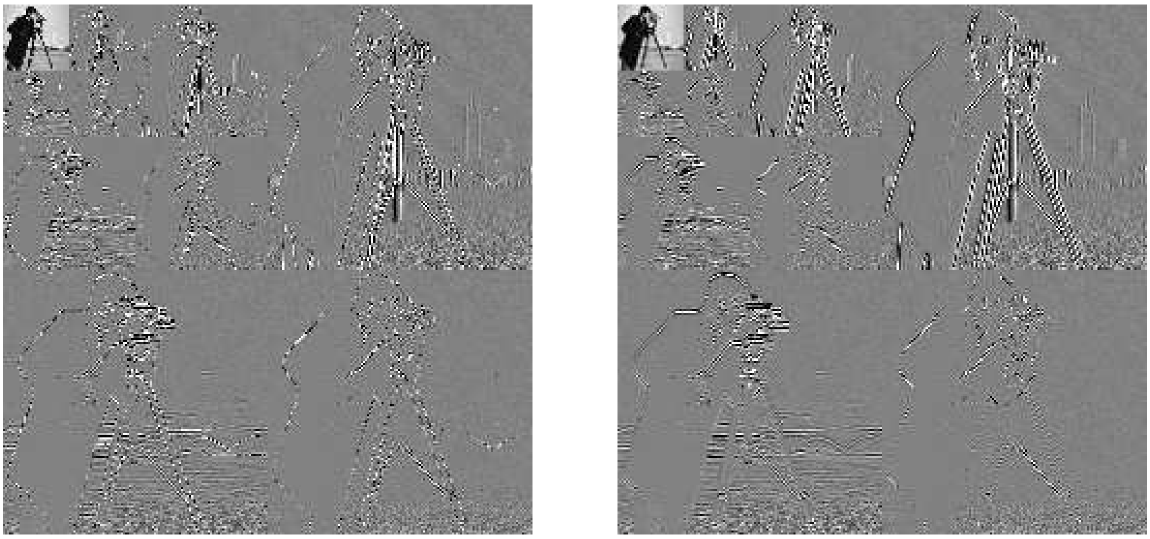


Figure 5.14: Subband analysis: region-based (left) and conventional (right)

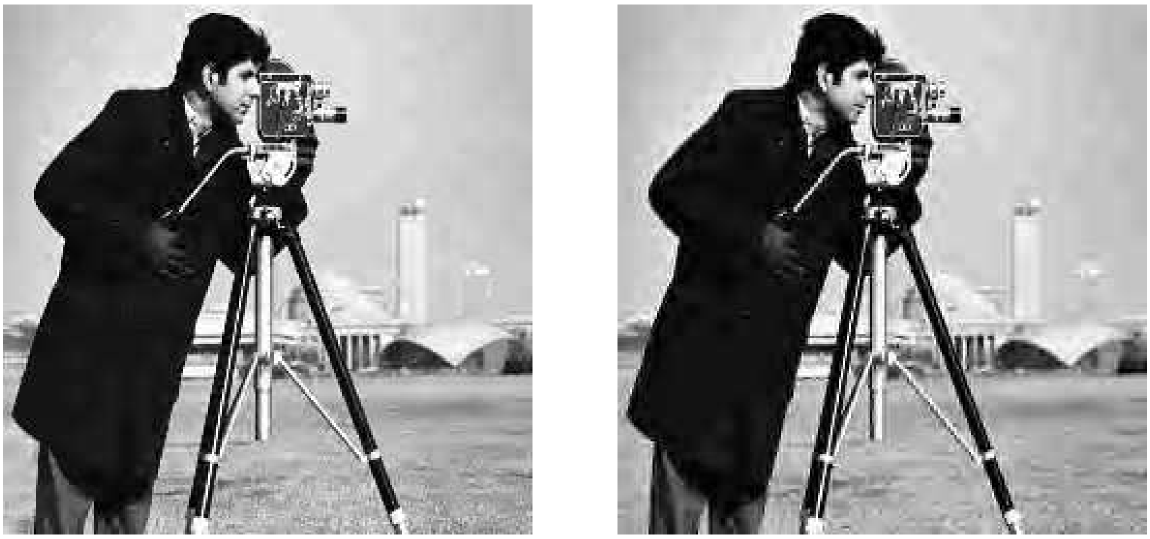


Figure 5.15: Subband coding results at 0.5 bpp: region-based (left) and conventional (right)

images are shown in Fig. 5.15. The whole bit rate is devoted to the subband contents in the conventional subband coding example. Its distribution among the different subbands is given in table 5.2. In the region-based example, only about 0.41 bits per pixel can be allocated to the subbands, because the rest is devoted to the overhead data: contour coding and quantizer choices. Table 5.3 gives the actual bit-rate values allocated for texture and overhead data. The distribution among the subbands varies according to the frequency contents of each region and also spatially among the regions.

In the conventional subband coding reconstruction, both ringing and blurring are present

Table 5.2: Subband rates in bits per pixel (bpp) for the conventional case

TARGET RATE	0.5 bpp
bands V1, D1, H1	0.08 bpp (16%)
bands V2, D2, H2	0.18 bpp (37%)
bands V3, D3, H3	0.10 bpp (20%)
lowest band L3	0.13 bpp (27%)
PSNR	26.9 dB

Table 5.3: Bit allocation for the region-based subband coding case

TARGET RATE	0.5 bpp
quantizers choice	0.020 bpp (4%)
contour	0.077 bpp (15%)
texture	0.410 bpp (81%)
averaged frequency distribution:	
bands V1, D1, H1	44%
bands V2, D2, H2	21%
bands V3, D3, H3	21%
lowest band L3	14%
PSNR	29.5 dB

along the sharp transitions and in high frequency details respectively. These effects are less visible in the region-based coded reconstruction. It is interesting to notice the reduced ringing effects along the sharp transitions located at the boundaries of the regions of the label image. Otherwise, either blurring or ringing are present, as happens along the left contour of the highest building in the background, which does not coincide with a transition between two regions. The overall quality of the region-based subband coding reconstruction is superior both from the point of view of the subjective perception and the objective PSNR measure. The fact that the filtering takes place inside relatively homogeneous regions, has been one of the main reasons for the observed improvements. Since filtering across sharp edges is avoided, the high-pass subbands are expected to contain less energy than for the conventional subband coding scheme. Furthermore, quantization errors do not influence neighboring regions, so that the ringing artifacts near sharp edges will be much less. Finally, the adaptive capability over spatial regions of the bit allocation algorithm is decisive to favor the important regions of the image, i.e. those regions with complex texture contents.

5.4.3 Comparison with other texture coding techniques

High compression requirements for images containing complex information contents lead to significant losses in all known image coding schemes [46]. In such cases, the simplification of the image resulting from a segmentation-based model for image coding is better tolerated than less natural artifacts, as ringing, blockiness and blurring, resulting from waveform image

Table 5.4: Rates and PSNR figures for the compared techniques at 0.5 bpp

RBSBC			SADCT		
18 regions	0.50 bpp	29.5 dB	14 regions	0.53 bpp	26.6 dB
RBDCT			RBWVD		
192 regions	0.52 bpp	26.2 dB	23 regions	0.56 bpp	26.7 dB

Table 5.5: Rates and PSNR figures for the compared techniques at 0.35 bpp

RBSBC			SADCT		
13 regions	0.35 bpp	25.7 dB	10 regions	0.37 bpp	22.5 dB
RBDCT			RBWVD		
110 regions	0.36 bpp	25.2 dB	28 regions	0.37 bpp	24.3 dB

coding techniques. This is due to the best matching of the image model –consisting of a partition into regions– to the perception characteristics of the visual system. From this point of view, the previous comparison of conventional subband coding versus region-based subband coding is not very fair.

In this section, the performance of region-based subband coding will be compared with other proposed texture coding techniques for region-based coding. Two target rates of 0.5 and 0.35 bpp have been selected for the experiments. The results are shown in Figs. 5.16 and 5.17. The actual rates and PSNR values are given in tables 5.4 and 5.5. The techniques are used in the same segmentation-based coding scheme [22].

Each texture coding technique is able to code efficiently the textures up to a given degree of complexity. Each one has a different amount of overhead information as well due to the coding of adaptive parameters for each region. The higher level optimization algorithm selects the best trade-off between texture and overhead information (contour and adaptive parameters) on a rate-distortion basis. Therefore, the optimal number of regions yielded by

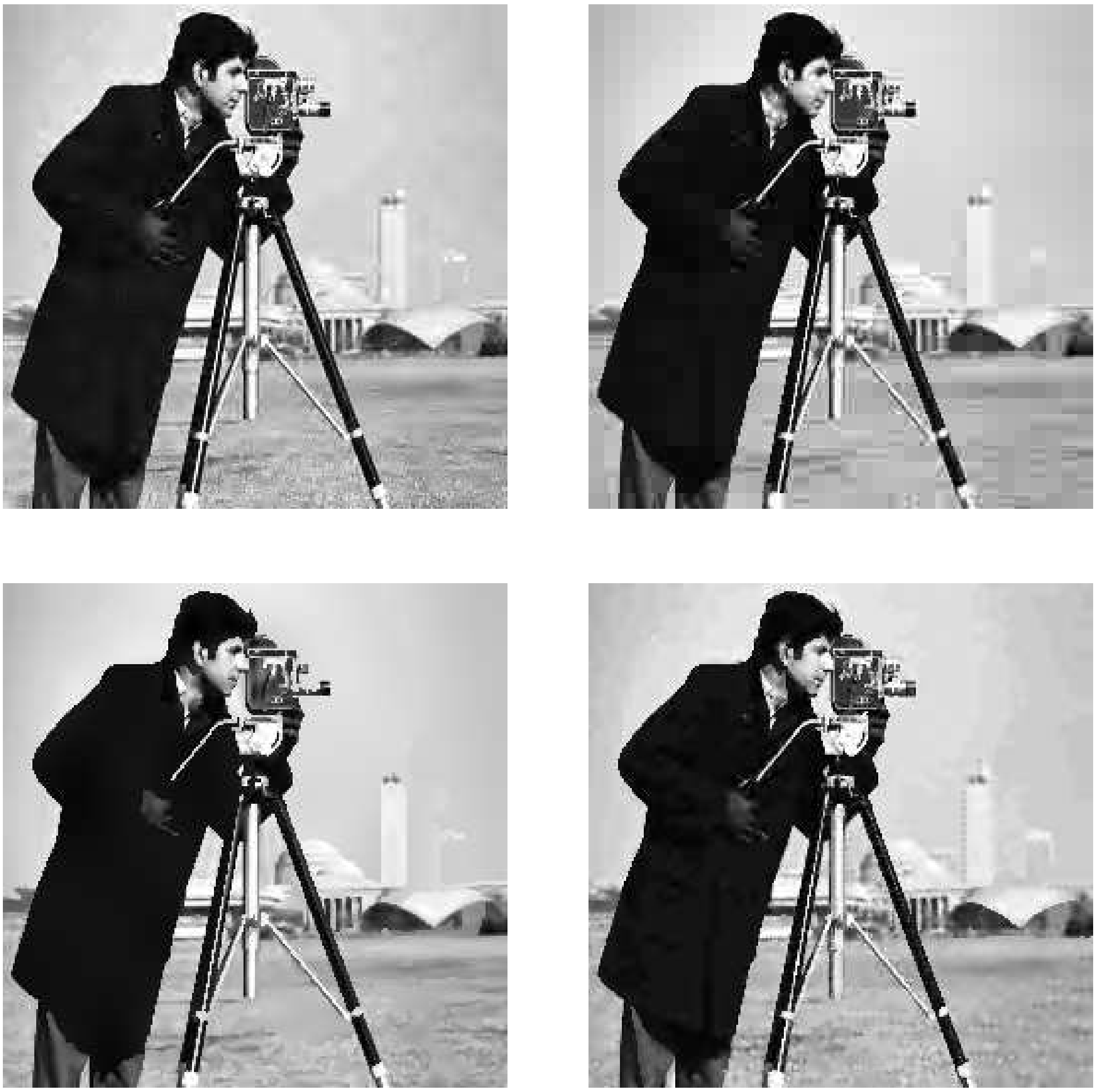


Figure 5.16: Performance comparison at 0.5 bpp: region-based subband coding (upper left), shape-adaptive DCT (upper right), region-based DCT (lower left) and region-based discrete wavelet transform (lower right)

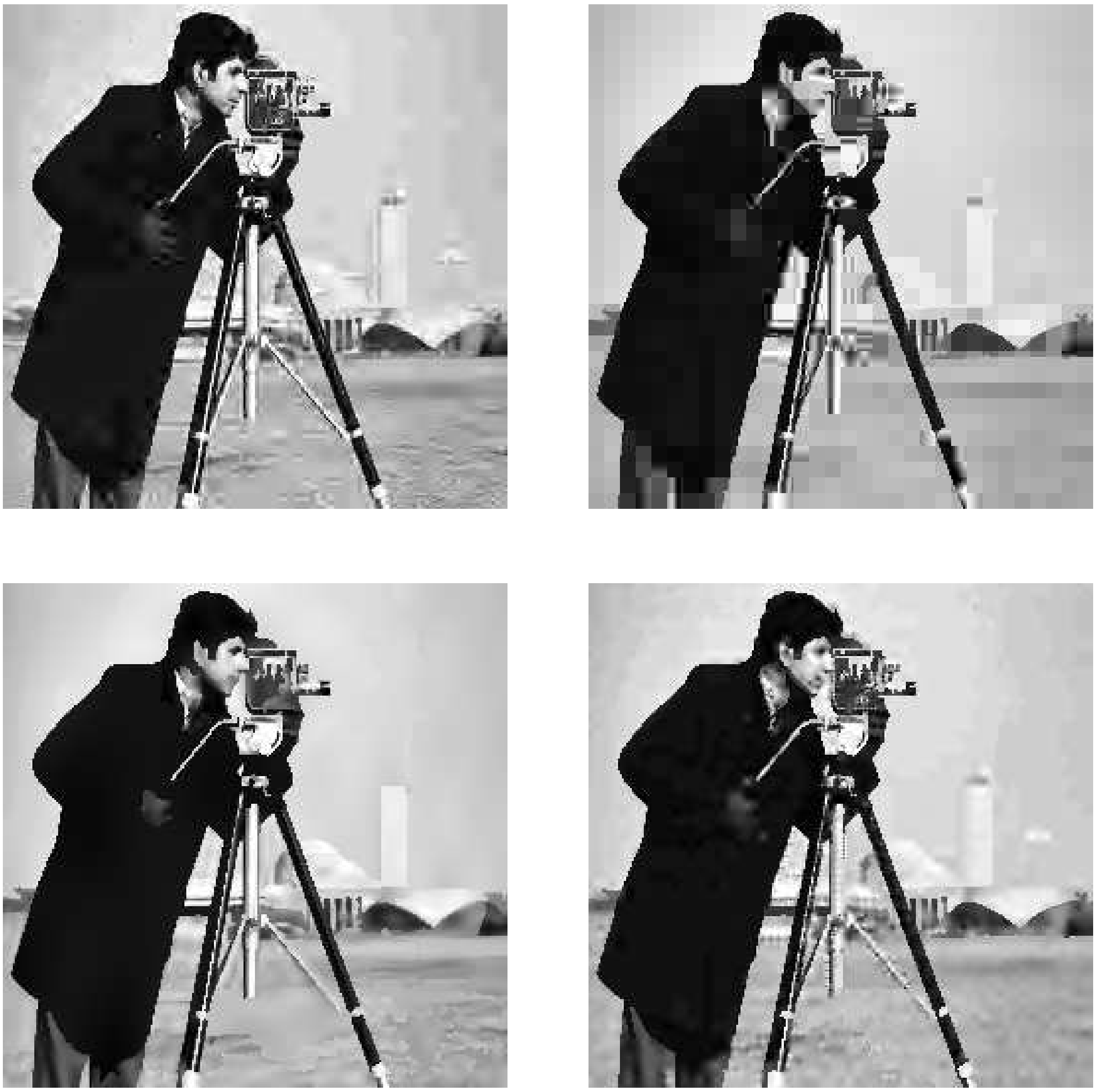


Figure 5.17: Performance comparison at 0.35 bpp: region-based subband coding (upper left), shape-adaptive DCT (upper right), region-based DCT (lower left) and region-based discrete wavelet transform (lower right)

the optimization algorithm for a given rate may be different for each one of the presented techniques. Except for the third technique, the ratio texture/overhead is between 75%/25% and 85%/15% for all of them. Let us make some remarks on the performances of each technique.

Region-Based Subband Coding (RBSBC): The results of the technique presented in the current work will be compared with the three reference techniques described below. Regarding the new result at 0.35 bpp shown in the upper left image of Fig. 5.17, notice the graceful degradation of both textures and sharp transitions with the decreasing bit-rate.

Shape Adaptive DCT (SADCT): This technique has been reported by Sikora and Makai in [107]. They apply the basic DCT coding scheme to square blocks which are completely included in the interior of the region, whereas for blocks crossing the region borders, a particular orthogonal set of separable DCT basis is used. At the highest bit-rate, it presents block artifacts in homogeneous textures, as can be observed in the grass. Some noise is also present in the proximity of sharp transitions. At the lowest bit-rate, for such a complex image, the blockiness is visible almost everywhere.

Region-Based DCT (RBDCT): The second reference technique is due to Gilge and has been reported in [37]. A set of cosine functions, orthogonal with respect to the shape of each region, is generated using Gram-Schmidt orthogonalization. The drawback of this method is the high computational load required for the generation of a large number of orthogonal basis for large regions. Therefore, only a set of 25 basis functions have been generated performing a least squares approximation of the complete set. This reduced set of functions is able to reconstruct only smooth textures and, thus, a finer segmentation is needed in order to obtain an acceptable result. In the lower right image of Fig. 5.16, the contour information is about 43% of the total. For the lowest bit-rate example shown in Fig. 5.17, it is about 50%. Please notice the poor rendition of the fine textures of the grass. There are some missing objects as well. For instance, one of the buildings of the background, the antennas in the highest building or some details of the camera are missing. These objects have not been properly segmented and the texture coding technique could not reconstruct them correctly as textures inside the regions.

Region-Based Wavelet Decomposition (RBWVD): This technique has been reported in [22]. The main differences with the region-based discrete wavelet transform introduced by Barnard in [5] is the fact that the filters employed here are non-separable 2-D wavelet basis. The drawbacks of wavelet basis filters regarding subband decompositions have been pointed out in appendix B. The non-linear phase characteristics of these operators and the regularity of the frequency response produce local artifacts along the edges in form of blurring. This can be observed in the walls of the highest building, which do not coincide with region transitions. A different kind of artifacts, due to lack of high frequency information, can be observed in the fine textures of the grass.

5.5 Analysis of the results of region–based subband coding

The results shown in Figs. 5.16 and 5.17 assess the superior reconstruction quality of the region-based subband coding scheme with respect to different techniques that have been proposed for texture coding. Although the results shown in this chapter have been only on the *cameraman*, similar results are obtained with other images. Chapter 6 will show results with different images in a complete perceptual segmentation-based model. Let us outline in this section the main conclusions that can be drawn from these results.

- The performance of the selected filter bank and the structure of the 10-band pyramid decomposition have proven to be good choices, as discussed throughout this chapter.

- The actual adaptation to the region contents and the ability to extract redundant information of the RBSBC scheme is better than for block-based coding, as the SADCT scheme, because the units of data to be coded (the regions) are more suited to the actual image contents than blocks.

- RBSBC has a smaller computational load than RBDCT through orthogonal basis. The superior performance of RBSBC with respect to the high frequency information inside the regions, as fine textures and details, has been clearly shown.

- Contrary to the problems for the coding of high frequency information of the RBWVD scheme, the rendering of the sharp transitions and highly textured areas is improved by RBSBC because of a better approximation of the perfect reconstruction property in presence of coarse quantization of the subbands.

- The results of the RBSBC scheme show significant objective and subjective (perceptual) improvements with respect to the current techniques normally used for region-based texture coding. These improvements hold for a wide range of bit-rates and for different types of images. Fig. 5.18 shows the reconstructed images at different target rates for both the original images *cameraman* and *Lenna*. The extension of RBSBC to the coding of color images and inter-frame images in video sequences (motion-compensated prediction error images) will be presented in chapter 6.

Table 5.6: Rates and PSNR figures for RBSBC at different target rates (in brackets)

CAMMAN (0.80)			CAMMAN (0.25)		
30 regions	0.83 bpp	32.7 dB	26 regions	0.27 bpp	24.6 dB
LENNA (0.50)			LENNA (0.20)		
33 regions	0.46 bpp	33.2 dB	18 regions	0.12 bpp	26.8 dB

- Nevertheless, region-based subband coding still shows significant drawbacks, as can be observed from the presented results. There are visible ringing effects in the neighborhood of sharp image transitions and bad rendition of small details. The more complex perceptual image model that has been proposed in chapter 2 complements the region-based subband coding technique aiming at the solution of such problems.

We would like to highlight the novelty of the presented region-based subband coding scheme. Related works in this area are those of Kwon and Chellappa [55] and Barnard et al [5]. The work of Kwon and Chellappa does not start from a segmentation of the original image. Instead, they realize a statistical analysis of the energy in the high frequency subbands in order to find edge features and perform spatially adaptive quantization. Concerning the work of Barnard et al, they employ a region-based discrete wavelet transform but do not optimize the segmentation, which is performed manually. The results of the RBWVD method presented here [22] are similar to those reported in [5]. An important difference is the analysis filter bank. First, because we use QMF's instead of wavelets and, second, for the symmetric extension technique. The variation to the original idea of Barnard [4] for the symmetric extension in the low pass-band that has been presented in section 5.2.4 results in a significant reduction of ringing effects for the reasons explained therein. Barnard himself observes an 'unexpected higher variance' [5, section 4, p. 1237] of the high-pass subbands, attributed to the fact that his extension method 'is not preserving the statistics' of the subbands. We have shown in Fig. 5.14 that the proposed extension method effectively reduces the amplitudes of the transition pixels in the high-pass subbands, being the responsible for better performance of the presented region-based scheme.

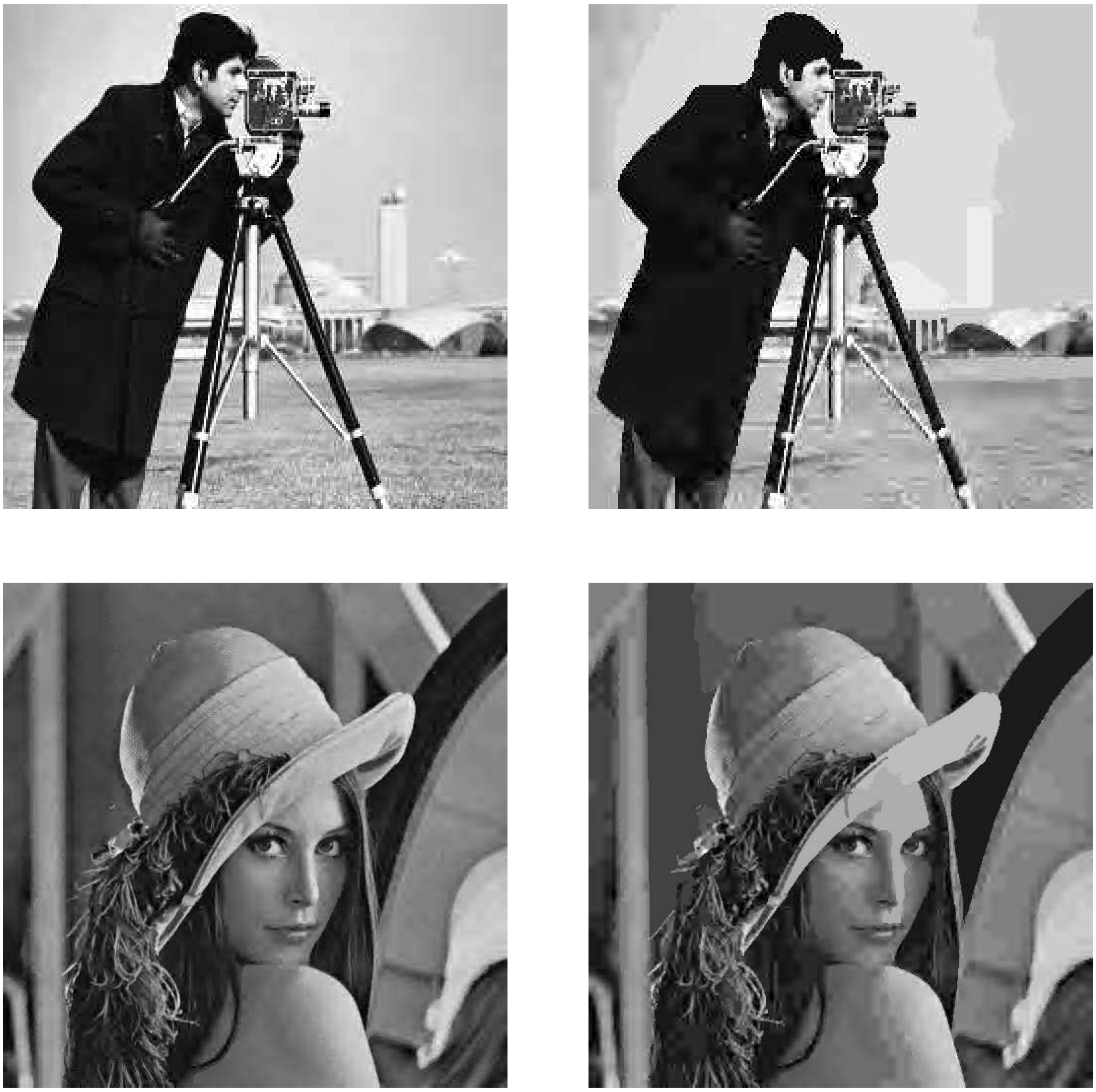


Figure 5.18: RBSBC reconstructed images at different rates. First row, *cameraman* at 0.83 and 0.27 bpp. Second row, *Lenna* at 0.46 and 0.12 bpp

Chapter 6

Application to image and video coding

Three perceptually motivated texture coding techniques have been investigated in preceding chapters, namely:

- a morphological interpolation technique for the reconstruction of smooth textures with application to sketch-based image coding schemes
- the extraction, selection and coding of small visual details taking into account their perceptual significance for the human visual system,
- a waveform image coding technique (subband coding) which has been adapted for the coding of homogeneous textures in a region-oriented model of the image.

These techniques are aimed at high compression ratios. This is the case of sketch-based image representations, or the individual coding of small visual features when a severe selection is necessary, or also the case of the representation of homogeneous textures in the framework of region-oriented image models. It is worthwhile to notice that the three techniques make use of structural primitives of visual signals. In particular, sharp image transitions are represented explicitly as edges, detail labels or partition contours.

These techniques have been independently shown. In this chapter, they will be combined together as the different components of a perceptual image model. First, the results of the application to still image coding are presented. In Section 6.1, the three coding techniques are combined gradually, following the heuristic reasons that led us to the particular image

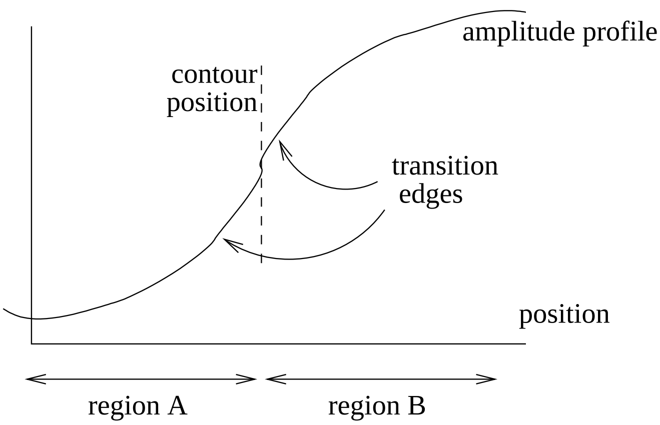


Figure 6.1: Amplitude profile across a sharp transition between two regions A and B

model that is proposed. Section 6.2 discusses the application of the texture coding techniques to video sequences, both in intra and in inter-frame mode.

6.1 Perceptual coding of still images

The results presented in chapter 5, show that the application of subband coding in the interior of the regions of a segmentation-based coding scheme, significantly improves the coding rendition with respect to the application of subband coding as a waveform coding technique to the whole image. Such improvement is due, on the one hand, to the adaptive distribution of the available rate among the regions depending on their texture contents and, on the other hand, to the decrease of ringing effects along the sharp transitions represented by region contours, because subband filters are applied separately on both sides of the transitions.

Nevertheless, the ringing effects have not been completely eliminated using the region-based subband coding scheme. These effects are still visible in the coded *cameraman* images at low bit-rates presented in Fig. 5.17 (upper left) and Fig. 5.18 (upper right). The reason for such ringing effects is explained in the following with the aid of the drawing of Fig. 6.1. The drawing represents the amplitude profile of a perpendicular section across an imaginary transition of the image.

Let us assume that the transition represented in the drawing is described by a contour of the partition separating the labels A and B. Subband filters perform separate frequency analysis and synthesis within the regions A and B on both sides of the transition. The texture contents of the regions are assumed to be homogeneous. However, as the label contours have zero width, the *edges* of the transition are included either inside region A or inside region B. This produces high frequency components of the analyzed texture of the region in the proximity of the contour position. The coarse quantization of such high frequencies is mainly responsible for the ringing effects.

6.1.1 Introducing the strong edge component

In order to avoid the ringing effects, the amplitude profile of the sharp transitions and the texture contents of the regions should be coded separately. This leads to a two-component image model with a *strong edge* component representing sharp transitions (from which the smooth areas can be interpolated) and a residual *texture* component, as proposed in most sketch-based coding schemes.

Open contours approach

Fig. 6.2 presents the coded strong edge component and the residual texture component (not coded). The strong edge component on the left side was already presented in Fig. 3.16 of chapter 3 (p. 60), and it is repeated here for convenience. It has been obtained by morphological interpolation from the upper and lower edge brims defined as the lines of largest curvature. The structural primitives of the image model are the positions and amplitudes of a set of 'open contour lines' (the sketch data). The coded sketch data represents a cost in bits per pixel of 0.17 bpp.

The residual image presented on the right hand side of Fig. 6.2 is simply the difference between the original image and the strong edge component. It contains the texture information, some errors in the neighborhood of the transitions and a number of small details. Conventional subband coding (without region segmentation) can be applied for the coding of this texture component. Fig. 6.3 (left) shows the subband analysis of the texture component. Please compare this frequency decomposition with the ones presented in Fig. 5.14 (on p. 115). The subband decomposition of the texture component does not present the high frequency information produced by the transitions. Even compared with the case of the region-based subband decomposition (Fig. 5.14, left) such high frequency information has been clearly reduced.

The information available from the strong edge component may be exploited for the coding of the textures. In this example, a quantization mask has been generated taking into account the position of the sharp transitions that have been coded for the strong edge component.

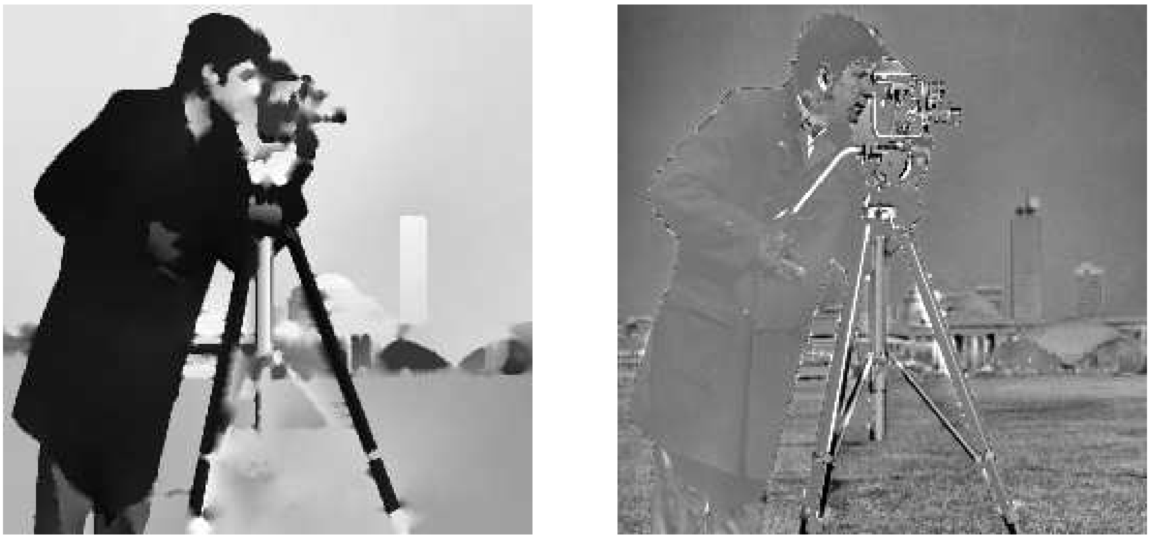


Figure 6.2: Strong edge component (0.17 bpp) and residual texture (not coded)

This mask is shown in Fig. 6.3 (right). The quantization mask allows a coarse quantization of the high frequency subbands at these positions, avoiding the disturbance produced by the errors along the transitions. Such errors are due to the approximated representation performed by means of the strong edge component, and appear as texture information in the residual image. However, as these errors are masked by the transitions themselves, they do not need to be accurately represented in the texture component.

The left image of Fig. 6.4 presents the texture component coded at 0.15 bpp. Observe that at such low bit-rate, only some smooth textures have been correctly represented. The details missing in the strong edge component have not been very well coded either. The result of adding the coded texture component to the strong edge component of Fig. 6.2 (left) is shown in the right image. The total rate for the addition of the two components is 0.17+0.15=0.32 bpp. This result can be compared with the results of conventional subband coding at 0.5 bpp and region-based subband coding at 0.35 bpp that were presented in the previous chapter. These results are shown again in Fig. 6.5 for an easier comparison.

The advantages of the separate coding of strong edges and residual textures in a two-component image model can be observed in the almost complete elimination of the ringing effects along the transitions that have been correctly represented in the strong edge component. However, with respect to the region-based coding result of Fig. 6.5 the rendition of the textured areas is poorer, such as in the area of the grass. This is due to the fact that the bit

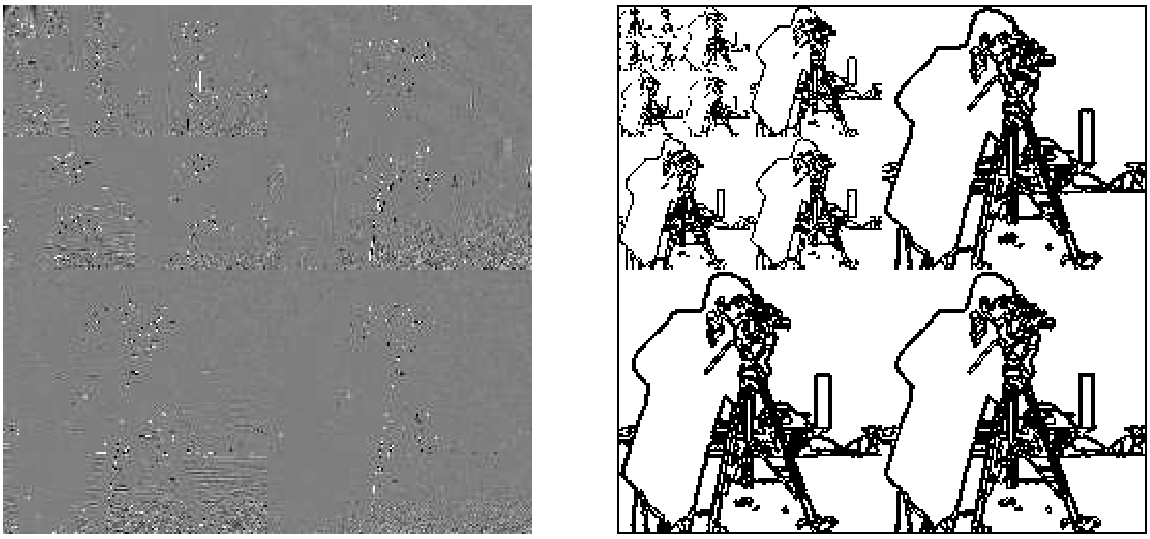


Figure 6.3: Subband analysis of the texture component and quantization mask

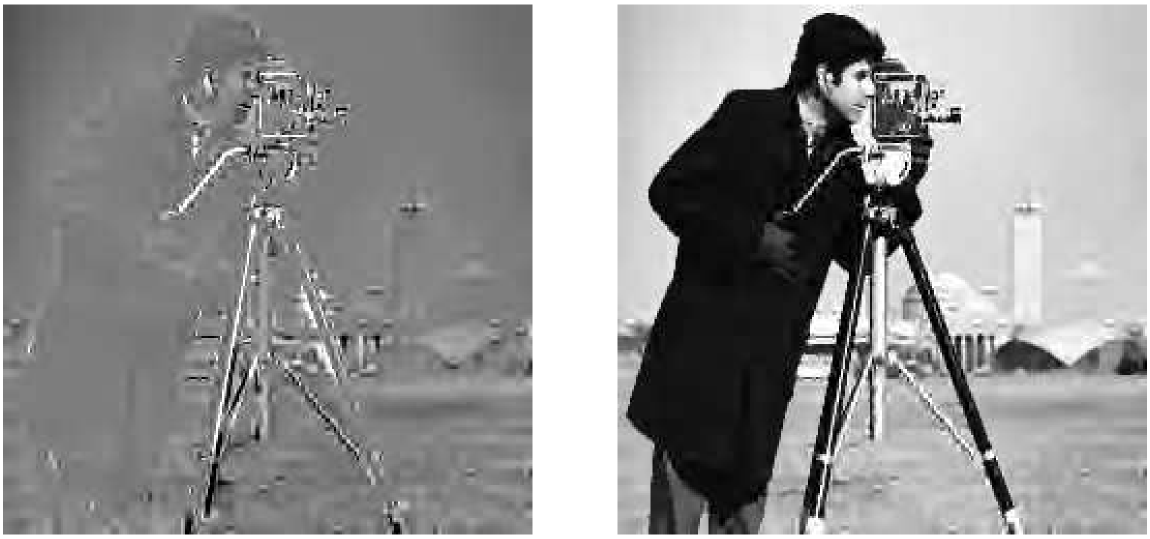


Figure 6.4: Subband coding of the texture component (0.15 bpp) and two-component reconstruction at 0.32 bpp

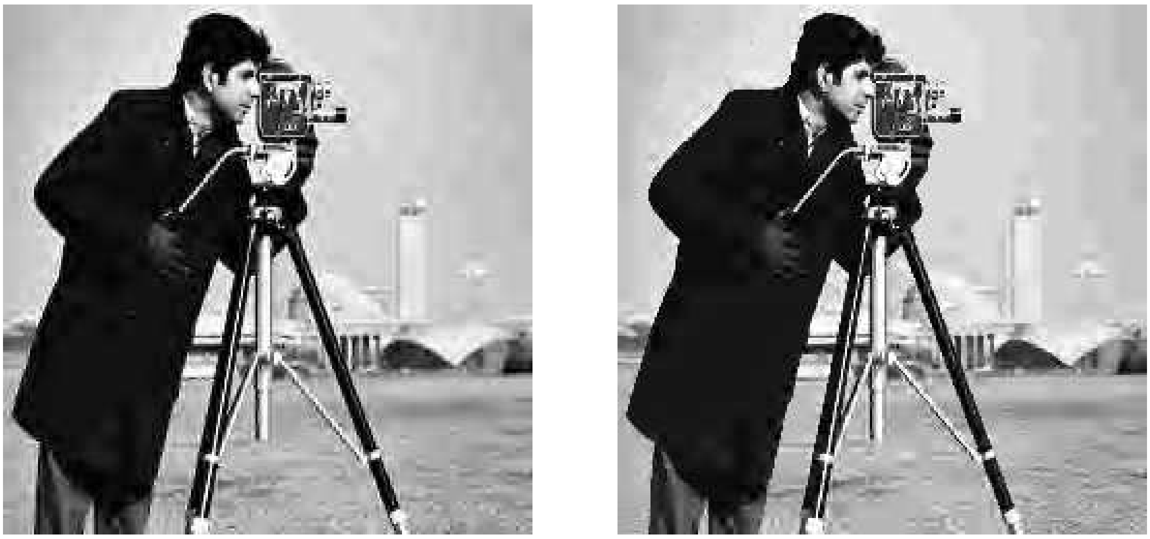


Figure 6.5: Conventional subband coding at 0.5 bpp and region-based subband coding at 0.35 bpp (from the previous chapter)

allocation for the subbands of the texture component cannot be performed adaptively as it was in the region-based coding scheme. The representation of strong edges by open contour features (edge brims) in the sketch-based model lacks of the 'watertight' structure of a division into mutually exclusive spatial regions provided by the segmentation process. Therefore, the independent assignment of bits to different regions of the image cannot be done and –except for the differences in the quantization step near the strong edges– all the areas of the image are equally considered by the subband coder.

This observation suggests that, in the context of coding, a description of the sharp transitions by means of closed contour features may be more useful. Most of the closed contours of a partition into regions give also a perceptual representation of the image. Even if some of the coded contours are 'false contours' –in the sense that they do not represent real image transitions but are necessary to close some regions– the benefits of a partition by means of regions with closed contours may compensate the coding of such false contours from the point of view of coding efficiency.

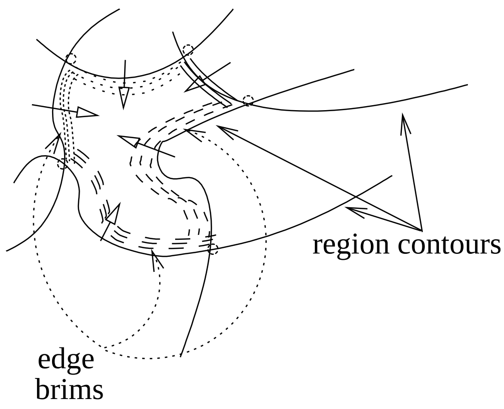


Figure 6.6: Interpolation of region contents from the amplitudes at the boundary pixels

Closed contours approach (region-oriented)

The example presented in section 2.2 of chapter 2 has introduced the possibility of a sketch-based representation of the strong edges in a region-oriented framework. In this example, edge brims are defined at the region contours. More precisely, the edge brims are composed of the pixels located on both sides of the contours separating different regions. This allows a smooth reconstruction of the textures of the interior of the regions by interpolation from the amplitudes of the pixels located at the region boundaries. The drawing on Fig. 6.6 illustrates this coding strategy.

The two sides of each contour separating two regions define the edge brims used for the interpolation of the interiors of the region. These brims will be called *region brims* in the following. Region brims span between the positions of two triple points (represented by small circles in the drawing of Fig. 6.6). The estimate of the amplitudes of region brims is made from the pixels of the region located at a certain distance (2 or less) from the boundary. The amplitude average of the region is computed and the difference between the pixel values of each brim and this average is approximated by a second order cosine function.

An example of texture coding by interpolation from region brims is shown in Fig. 6.7. This image is taken now as the strong edge component of the model. The resulting bit-rate is 0.15 bpp, including the partition information. The segmentation technique is the same that was explained in chapter 5 for the region-based subband coding technique.

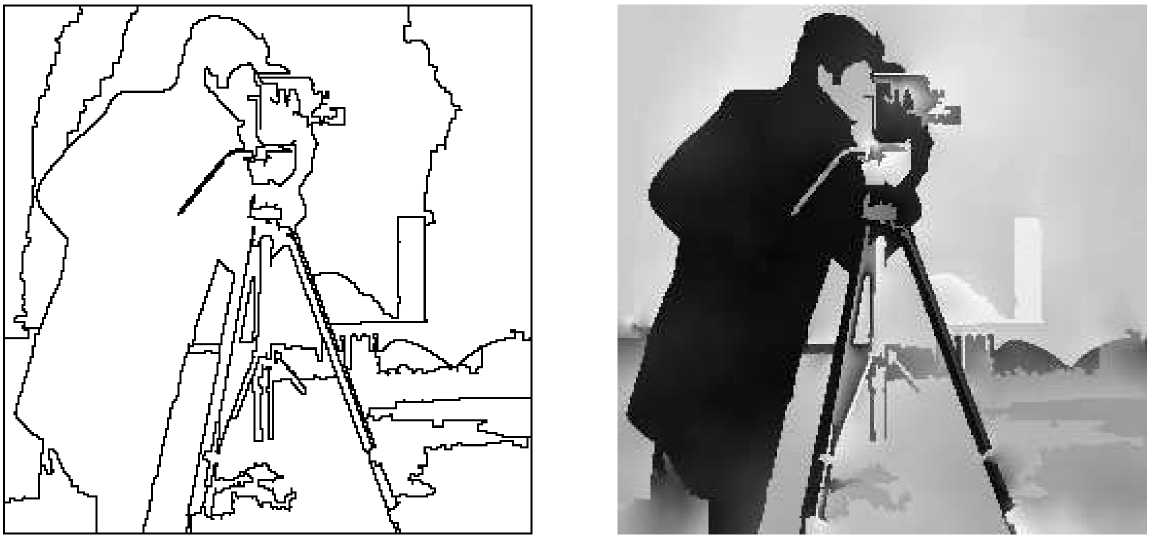


Figure 6.7: Partition of the cameraman image (31 regions) and strong edge component interpolated from ‘region brims’ (0.15 bpp)

6.1.2 Introducing the details component

The strong edge component shown in Fig. 6.7 could be used as a coarse approximation of the image for high compression applications. However, the lack of some significant details gives an unnatural aspect to this image. In order to improve the coding rendition of the strong edge component there exist several strategies:

- the segmentation may be refined until we get the desired precision in the representation of the objects by introducing new regions
- the residual texture may be coded, using a given texture coding technique, in order to represent the residual information
- a purpose-designed technique may be applied to obtain the features of interest.

The strategy of refining the segmentation often consists in increasing the degree of homogeneity defined in the segmentation process for the extraction of the regions. The more rigid the homogeneity criterion, the larger the number of resulting regions. However, the possibility of getting an over-segmented partition is not desirable, given the cost of the coding of a large number of contours.

A more convenient strategy would be the second one. Those objects whose contours are not represented in the partition, are coded as textures inside the segmented regions. However, texture coding techniques assume that the interiors of the regions are homogeneous –smooth or with periodical or somehow repeated texture patterns– and the non-extracted contours are seen as spurious transitions. This will result in a poor coding rendition at the contours of such objects. This is even worse if the object is a blob of small size. This effect can be observed, for instance, in the two-component reconstruction of Fig. 6.4. The antennae on top of the high building in the background, that were coded in the texture component as a 'texture' of the sky, have not been properly represented.

In some applications, the individual extraction of the 'missing objects' and the selection of the most significant ones by means of a consistent criterion can give better results. The detail coding technique explained in chapter 4 is aimed at the extraction of small objects which are visually meaningful. The application of detail coding in this example is illustrated in Figs. 6.8–6.10. Fig. 6.8 (left) shows the coding residue of the strong edge component presented in Fig. 6.7. The details extracted from this residue are shown in the right image. Fig. 6.9 presents the result of the ranking and selection of the extracted details. As in the examples of detail ranking of chapter 4, the brighter the label, the more significant the detail. A budget of 2000 bits permits the selection of the first 25 ranked details for coding. The selected details are presented in the right image of Fig. 6.9 with the coded amplitude values.

Fig. 6.10 shows the reconstruction obtained with the strong edge component of Fig. 6.7 and the details component of Fig. 6.9. Please notice the perceptual significance of the coded details. The total bit-rate is 0.15+0.03=0.18 bpp, what yields a compression ratio of 45 for this coded image. The right image presented in Fig. 6.10 is the new coding residue of the two-component reconstruction.

6.1.3 Region-based subband coding of the residual textures

The fact that the coded details are removed from the coding residue makes easier the coding of the remaining homogeneous textures in the texture component. In addition, the partition structure of the region-oriented image model used in the strong edge component allows the coding of the different regions at different bit-rates. As explained in the previous chapter, a rate-constrained optimization is performed by a higher level optimization algorithm [69] for the subband coding of the textures of these regions. Furthermore, given the rate constraint and a set of quantizers (or texture coding techniques), the optimization algorithm can derive useful information in order to obtain the optimal segmentation from a set of possibilities that are analyzed in the partition tree. This rate-constrained optimization scheme is explained in more detail in appendix C.

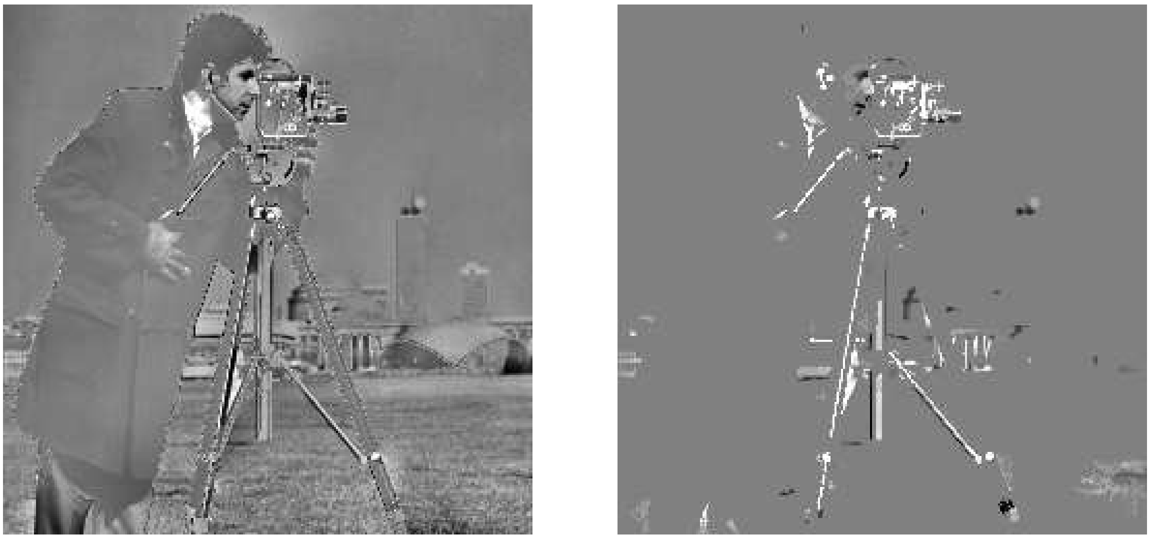


Figure 6.8: Coding residue of the strong edge component of Fig. 6.7 (right) and extracted details

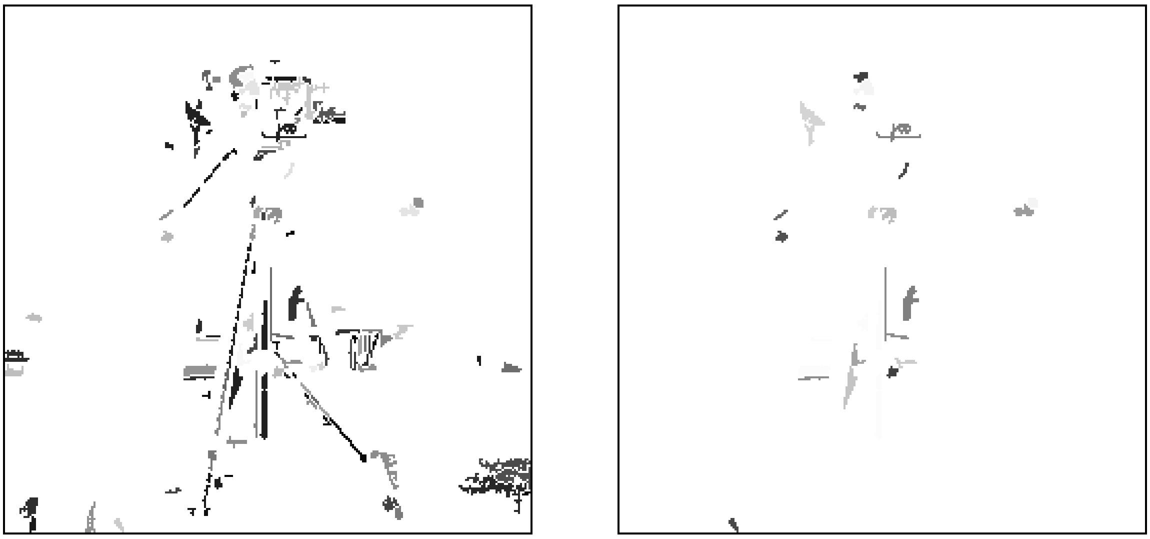


Figure 6.9: Ranking of the extracted details and 25 selected details coded with 2000 bits (0.03 bpp)

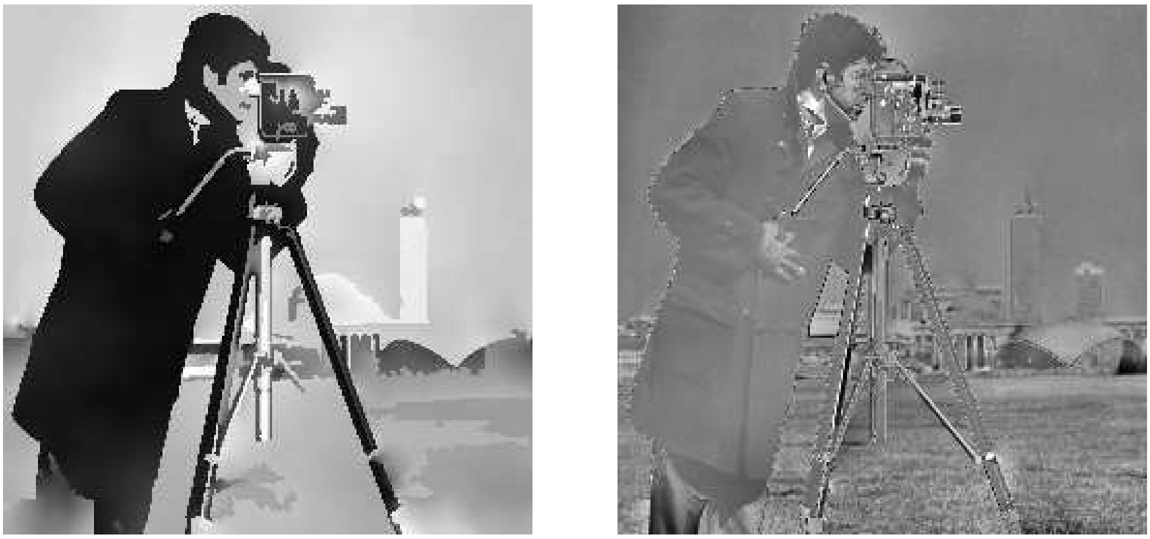


Figure 6.10: Two-component coding (strong edges + details) of the cameraman image at 0.18 bpp and coding residue

The residual texture of the right image of Fig. 6.10 is therefore coded by the region-based subband coding technique. The coded texture component at 0.15 bpp is shown in the left image of Fig. 6.11. This rate is devoted only to texture information. The partition information has been already coded in the strong edge component. The addition of the texture component to the previous two-component reconstruction (Fig. 6.10, left) yields the final result of Fig. 6.11. The total rate for this three-component coding is thus 0.33 bpp. The distribution of the rates among the three components is summarized in Table 6.1. This image can be compared with the result of region-based subband coding only presented in Fig. 6.5 (right). The buildings in the background or some details in the tripod, for instance, are better represented in this case. The three component image model yields quality improvements with respect to the ringing artifacts and the coding rendition of the small details.

6.1.4 Three-component coding of color images

An example of application of the three component model in the context of color images is shown in Fig. 6.12. An original frame from the *foreman* sequence is shown in the top left image. The partition into regions is presented in the top right. The extracted details have

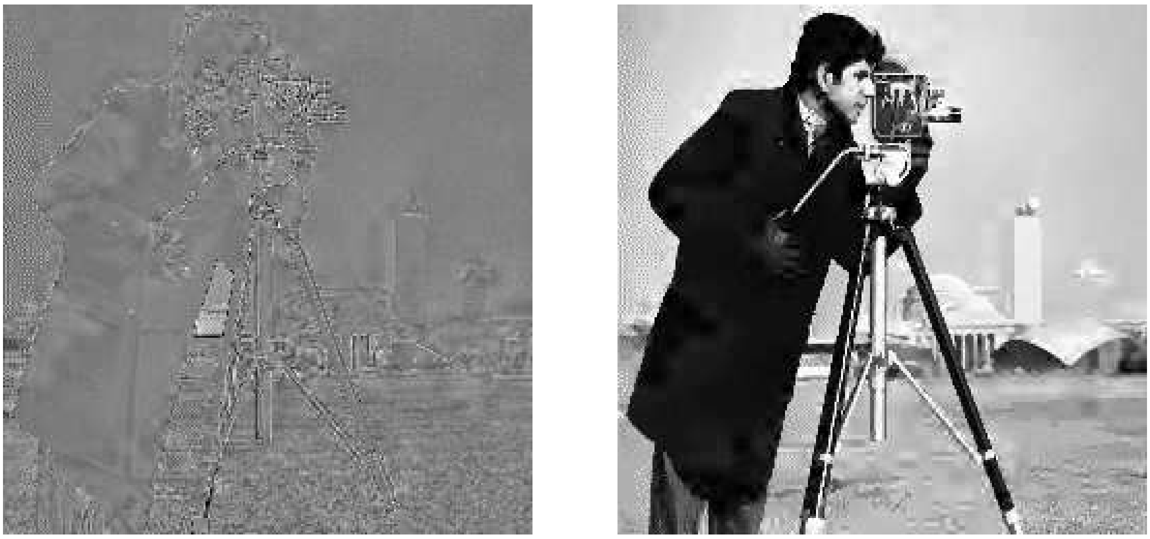


Figure 6.11: Subband coded textures at 0.15 bpp and three-component coding (strong edges + details + texture) at 0.33 bpp

Table 6.1: Component rates for the reconstruction of Fig. 6.11

Coded component	number of bits	component rate [bpp]
strong edges:		
amplitudes	2712	
partition	7200	
total	9912	0.15
details:		
amplitudes	248	
positions	1808	
total	2056	0.03
textures:		
quantizers	544	
subbands	9144	
total	9688	0.15
total rate:		0.33

been also included in this image. As the details are displayed with the coded colors, they can be clearly distinguished from the contours of the partition. There are 18 details that have been extracted, as in the previous example, from the coding residue of the strong edge component. The morphological operators for detail extraction described in chapter 4 have been applied to a weighted combination of the values of the luminance Y and the color signals U and V, so that not only luminance details but also salient color features are extracted.

The bottom left image of Fig. 6.12 displays the reconstruction obtained from the strong edge component and the details component. Observe that the smooth textures are mixed in some regions, such as the face. This is a drawback of the reconstruction from edge brims that occurs when some contours are not present in the partition. In this case, the piece of the contour corresponding to the right side of the man's face is missing. The amplitudes of the edge brims of the large region including the face and the shadowed area of the building are mixed by the morphological interpolation technique. The reconstructed amplitude values are a mixture of both areas, i. e. part of the building gets the redish color of the face and the face gets the dark color of the shadow. This effect can be partly reduced by the coding of the residue of this two-component reconstruction in the texture component.

The bottom right image of Fig. 6.12 shows the three-component reconstruction. It is obtained from the combination of the strong edge component, the details component and the coded textures (the texture component is not shown). The rate distribution among the coded components for this example is given in table 6.2. The total bit-rate in this case is 0.27 bpp.

The figures given in table 6.2 include the cost of the color information. The bits spent for the amplitudes of the color components are about 7% of the total bit-rate. Each texture component deals with the color information in a specific manner. The following list describes the particularities of each texture coding technique for the coding of color images:

- Strong edge component
  The amplitude values of the color signals U and V at the positions of the edge brims are represented with lower order approximation (first order functions) than luminance amplitudes. The smooth color information of each region is reconstructed by interpolation from these coded amplitudes. Morphological interpolation is independently applied for the reconstruction of the luminance signal Y and the color signals U and V. The interpolation algorithm is applied to the same set of region brims with the different coded amplitude values of Y, U and V.

- Details component
  In the details component, the averaged values of signals U and V inside each detail label are also more coarsely quantized than the luminance amplitudes. This may done given

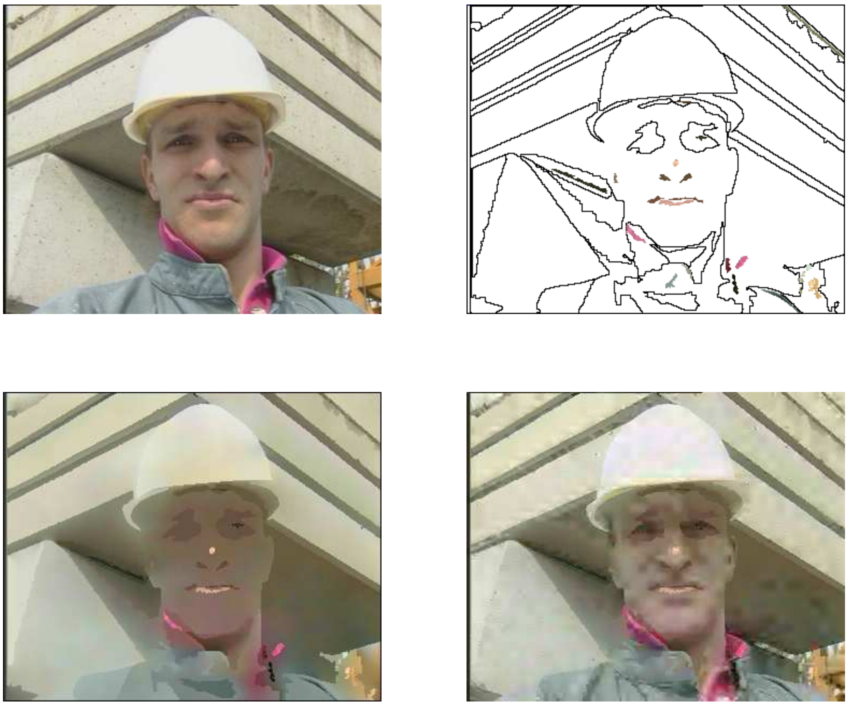


Figure 6.12: Three component coding of the *foreman* image: top left, original image; top right, regions of the partition and extracted details; bottom left, strong edge component and details; bottom right, three-component coding (strong edges + details + texture) at 0.12 + 0.04 + 0.11= 0.27 bpp

that the color information is not so accurately perceived for such small visual features [82].

- Texture component
  As indicated in section 5.3.3 when the design of perceptual quantizers for subband coding was discussed, the highest frequency bands corresponding to color signals may be simply discarded. In addition, quantization errors in color signals are given smaller weights than luminance quantization errors by the bit allocation algorithm. This results in a rate of the coded color texture component much smaller than the rate allocated to the luminance signal.

Table 6.2: Component rates for the reconstruction of Fig. 6.11

Coded component	number of bits	component rate [bpp]
strong edges:		
amplitudes	872	
partition	2152	
total	3024	0.12
details:		
amplitudes	192	
positions	928	
total	1120	0.04
textures:		
quantizers	488	
subbands	2328	
total	2816	0.11
total rate:		0.27

6.2 Application to video sequences

The main difference in the coding of video sequences with respect to the coding of still images is that video coders have to deal with *motion*. Scene motion may be described by simple spatial displacements of the structural primitives of the image model (such as blocks, regions or edges) or by more complex 2D or 3D motion models which are able to characterize not only translation displacements but also the deformations of the object projections in the camera plane produced by the motion of the objects in the 3D scene. Affine motion models, for instance, are able to describe tilt or zoom (motion in the z axis) whereas 3D motion models can describe complex movements such as rotation.

The analysis of scene motion (motion estimation) permits the coding of most temporal variations in the video signal as spatial 'movements' of the contents of the previous frames. If the motion parameters are coded and transmitted to the receiver, the decoder is able to reconstruct the current frame from the information that was already coded in previous frames using these parameters (motion compensation). For most areas of the image, only some changes in luminance that cannot be accurately described by the motion parameters are coded as spatial information. This is known as *inter-frame* coding mode, and consists in the coding of the difference between the original frame and its motion-compensated reconstruction.

On the other hand, the areas of the image that cannot be predicted from the information available in previous frames have to be coded separately. This happens, for instance, for the whole image in the first frame of the video sequence or when a scene change occurs. In these cases, the whole frame is coded as a still image in *intra-frame* coding mode. Intra-frame coding is also applied in some areas of other frames. Typical situations where intra-frame coding is required are, for example, the newly appearing objects in the scene or the backgrounds uncovered by the moving objects.

In this thesis, we have not studied the problems related to motion in video sequences. However, the segmentation-based video coding scheme employed for the application of the texture coding techniques [22] performs motion estimation and compensation using an affine model of the motion of the segmented regions. The projection step (see appendix C) tracks the evolution of the regions along the time and the textures of the regions that can be correctly predicted from the previous frames are coded in inter-frame mode. The textures of the new regions appearing in each frame or those regions where motion compensated coding is not efficient –in terms of rate-distortion– are coded in intra-frame mode.

Therefore, this section deals with the application of the investigated texture coding techniques to the coding of region textures in video sequences. For the case of intra-frame coding, these techniques and the image model are applied as in the case of still images. Let us briefly discuss the differences in the application of the three texture coding techniques in the case of inter-frame coding.

Inter-frame region-based subband coding of textures

Region-based subband texture coding was already applied to the coding residue of the two first components of the model (strong edges and details) for still image coding. Therefore, its application to the coding residue (or prediction error) of the motion compensated region textures does not represent significant differences.

Inter-frame residual textures may present different statistical properties that will affect the quantization steps and the bit allocation among the subbands. However, as the region-based subband coding technique adapts these parameters to the contents of the coded region by means of the rate-constrained optimization algorithm, as explained in section 5.3 (p. 101), the only requirement is that such algorithm must be provided with an adequate set of quantizers that can match the properties of the inter-frame textures. Gharavi has proposed in [35] to use quantizers with the same characteristics that the one described in section 5.3.3 (Fig. 5.6).

From the experience obtained in the application to the inter-frame case, we have observed that the optimal quantizers for inter-frame textures tend to have larger dead-zones and smaller quantization steps. Therefore, we have extended the set of available quantizers to include the ones with appropriate characteristics for inter-frame coding.

Morphological interpolation and inter-frame coding

Morphological interpolation does not seem to be adequate, in principle, for inter-frame texture coding. This technique has been proposed for the coding of the strong edge component, containing the strong edges and smooth areas. Such texture contents are present in original images but not in residual textures. At the most, inter-frame prediction error images are almost zero in the regions where motion-compensation has successfully predicted the texture contents. The residual textures of these regions can be simply approximated by lower order smooth functions such as polynomials, cosines or even the average value.

The application of morphological interpolation for the coding of these textures from the residual values at the positions of the region brims has been considered as an alternative coding technique. Morphological interpolation yields a smooth reconstruction of such textures which is cost-efficient and able to compete successfully in terms of rate-distortion with the previously mentioned smooth functions for the coding of inter-frame textures. The reason for this result, is that motion-compensation prediction errors sometimes present edge-like features which are very localized at the boundaries of some regions. The reconstruction of these errors by interpolation from the nearest region brim produces a localized reconstruction of these residual features.

On the other hand, the application to inter-frame coding of a two-component model consisting of the strong edge and texture components has not given significant improvements from the point of view of the reconstruction PSNR. Nevertheless, we have observed that this two-component coding of inter-frame textures yields reconstructed images with smaller ringing artifacts. This can be explained, as before, because of the edge-like features of the motion-compensated prediction error in the boundaries of the regions.

Detail coding in video sequences

The detail coding technique can be applied for the coding of details in video sequences as explained in section 4.1.2 of chapter 4. Actually, the temporal persistence of a given detail through consecutive frames of the original sequence has been taken as a parameter to assess its perceptual significance.

A drawback of the detail coding technique is the temporal connectivity assumption that was made in order to track details along the time. This forces the extraction of such details frame by frame, even if temporal subsampling is applied for very low bit-rate video coding applications, with the added computational load of the extraction process. Details are extracted in all the frames of the original video sequence to allow the 'temporal marking' (see eq. 4.7 on p. 71), but the amplitudes and positions of these details are not coded in the skipped frames. However, even performing such tracking, fast moving details are not adequately tracked. This makes the detail coding technique efficient only in the case of smooth motion of the original sequences, for example in the typical head and shoulders images of video-conference applications.

6.2.1 Video coding results

Two original color video sequences in QCIF format have been chosen for the application of the texture coding techniques. The segmentation-based video coding scheme is explained in appendix C. For the sequence *foreman*, a target rate of 42 kbit/s and a frame rate of 5 Hz have been chosen. The coding scheme is applied for this sequence with the following texture coding techniques:

- in intra-frame mode:
    - morphological interpolation only (strong edge component)
    - region-based subband coding only
    - morphological interpolation and region-based subband coding (two-component model)

- in inter-frame mode:
    - morphological interpolation only
    - region-based subband coding only.

This original sequence presents significant motion. In particular, an important amount of camera motion occurs in the first frames because the man is handling the camera himself, then the camera pans to the right towards a building under construction.

As in the case of still image coding, the rate-constrained optimization algorithm decides both the partition structure for each frame and the texture coding technique applied for the coding of the regions. This decision is performed on the basis of the rates and distortion values resulting from the application, both in intra-frame and inter-frame mode, of each one

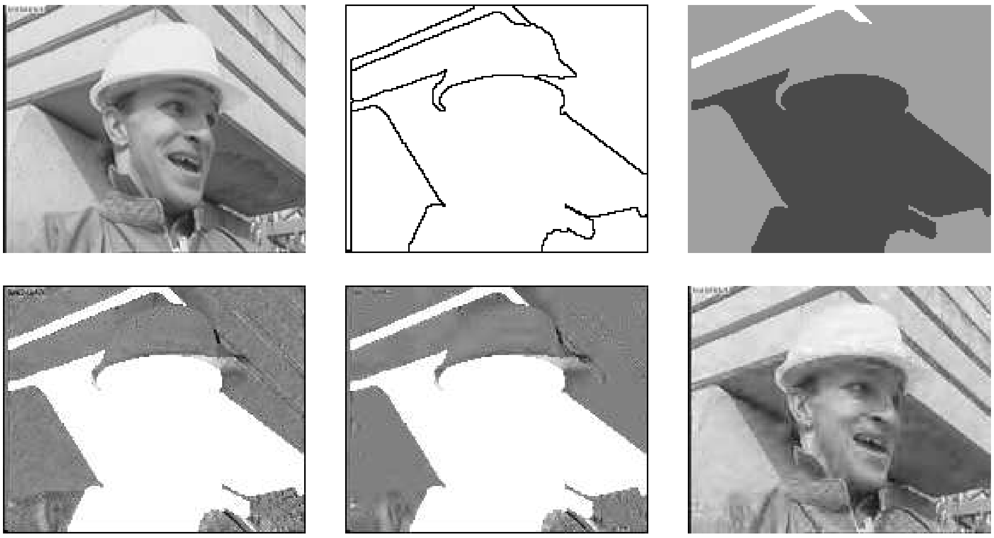


Figure 6.13: Results on video sequences: *foreman* (frame No. 132). First row: original, partition and decision map. Second row: prediction error in inter-frame regions, coded prediction error and coded frame (45 kbit/s)

of the texture coding techniques (with different possibilities of coding accuracy for the case of region-based subband coding) to every region of the partition proposals analyzed in the partition tree. The optimization algorithm selects the optimal partition proposal and the set of coding techniques that result in the smaller distortion for a given rate.

The coding results for two frames of the sequence *foreman* are presented in Figs. 6.13 and 6.14. The images in the first row are, respectively, the original frame, the partition and the decision map. The second row shows the prediction error in inter-frame regions, the coded prediction error and the final coded image, with the intra-frame coded regions and the inter-frame coded textures added to the motion-compensated prediction. The decision map indicates the texture coding technique employed in the coding of each region. Dark grey corresponds to the two-component coding (interpolation and subband), light grey, subband coding only and white is reserved for morphological interpolation only.

The decision map of frame No. 132 (Fig. 6.13, top right) indicates that the large region in the center of the image has been coded in intra-frame mode. In this frame, the man turns his head to the right with a sudden movement. The optimization algorithm has chosen to

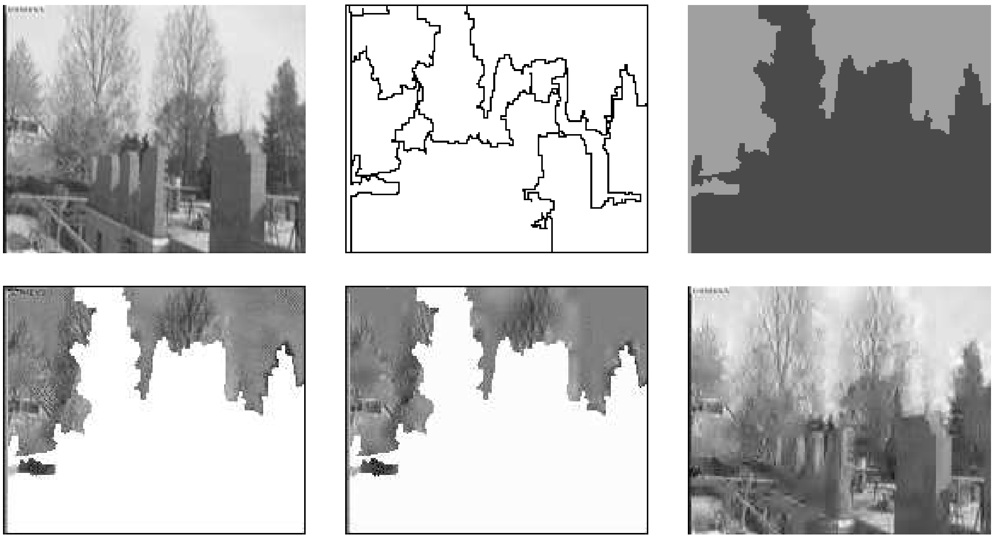


Figure 6.14: Results on video sequences: *foreman* (frame No. 216). First row: original, partition and decision map. Second row: prediction error in inter-frame regions, coded prediction error and coded frame (45 kbit/s)

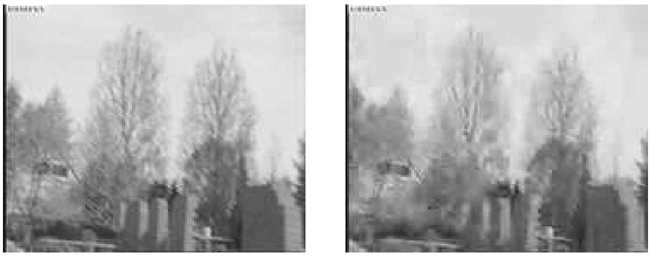


Figure 6.15: Previous frame (No. 210) for the motion compensation of the coded image in Fig. 6.14. Left: original. Right: coded

code this region in intra-frame. Most of the regions in intra-frame mode, are coded using the two-component model, with morphological interpolation for the region brims and subband coding of the residual textures. Except for one region of the building, the other regions are coded in inter-frame mode using region-based subband coding only, with different bit-rates for each one of the regions. For example, the region of the helmet is coded with compression 120, i. e. 0.066 bppr[1], whereas the other regions are coded with compressions ranging from 36 (the bottom right region) to 88 (the top right), 0.22 to 0.09 bppr.

The coded frame No. 216 in Fig. 6.14 has been chosen because of the large motion involved (this is the last one of the frames where the cammera pan occurs). The previous original and coded frames are displayed in Fig. 6.15 in order to show the motion involved. Recall that the motion compensated prediction is obtained from the coded frame on the right of Fig. 6.15. Almost all the regions of this frame were coded in intra mode. Please notice the the differences in the texture part of trees of the coded image in the bottom right of Fig. 6.14. In the area already coded in the previous frame and that is used for the motion-compensated prediction, the residual texture coding performed by the region-based subband coding technique in inter-frame mode improves the quality of the textures with respect to the part that is coded in intra-mode.

The time evolution of the bit-rate, PSNR values, number of regions and portion of pixels where region-based subband coding is applied are shown in the plots of Fig. 6.16 as a function of the frame number for the sequence *foreman*. The bit-rate, for instance, is allowed to be rather high for the first coded frame (15000 bits) and gradually tends to the targeted value. This rate is the total rate of the coded sequence, including partition information (20%), color and luminance texture information (73%), choice of the texture technique for each region

[1] The symbol *bppr* stands for bits per pixel of the region, as introduced in chapter 5

(1.5%) and motion parameters (5.5%).

The PSNR values are not constrained at all and present significant variations depending on the motion of the scene and the contents of each frame[2].

The initial number of regions is 27 and tends to decrease in the frames with slow motion, given that most of the regions can be motion compensated with similar motion parameters and this allows the merging of such regions. When a sudden motion occurs, as in the fast camera motion mentioned above starting at frame 150, the number of regions, that have decreased down to 5 regions at this point, progressively increases.

The last plot of Fig. 6.16 shows the portion of textures coded using the subband technique, i. e., either region-based subband coding only or two-component coding (morphological interpolation + subband) in intra-frame mode. Most of the textures coded by morphological interpolation only are motion prediction errors of regions with slow motion. Some regions containing smooth textures in the original image are also coded using the strong edge component only in intra-frame mode. However, the average portion of subband coded textures is above 68% of the total pixels of the sequence.

In the previous coding example using the *foreman* sequence the details component has not been introduced because the sequence presents too much motion. As explained above, this does not allow the correct extraction and tracking of the details along the time. If the details component had been coded with this sequence, this would have resulted in a lack of temporal stability of the extracted details, i. e. most of them would have not been kept coded from frame to frame and would appear and disappear in consecutive frames.

A second QCIF color sequence presenting less motion than the former, the *car-phone* sequence, is taken to show the application of the details component. In this case, the target bit-rate is 30 kbit/s, the frame rate is also 5 Hz and a certain number of details is selected for coding in each frame. The results are shown in Fig. 6.17. The images of this figure are ordered as before, except for the center image of the bottom row, where the extracted details have been superimposed to the coded residual textures. Notice that only one eye of the man is coded as a detail in this frame. This is due to the fact that, though it was extracted by the morphological operator, it is not ranked high enough to be selected. Actually, this detail is less contrasted than the ones that are selected from the residue of the strong edge component. Therefore, it is left for the the subband coding technique to be coded as a texture of the image.

[2] The optimization algorithm [69] allows to set a minimum threshold quality at the expense of not reaching the target compression in some complex frames. This threshold has been set to 20.0 dB in the current simulation, in order to obtain a smoother control of the bit-rate

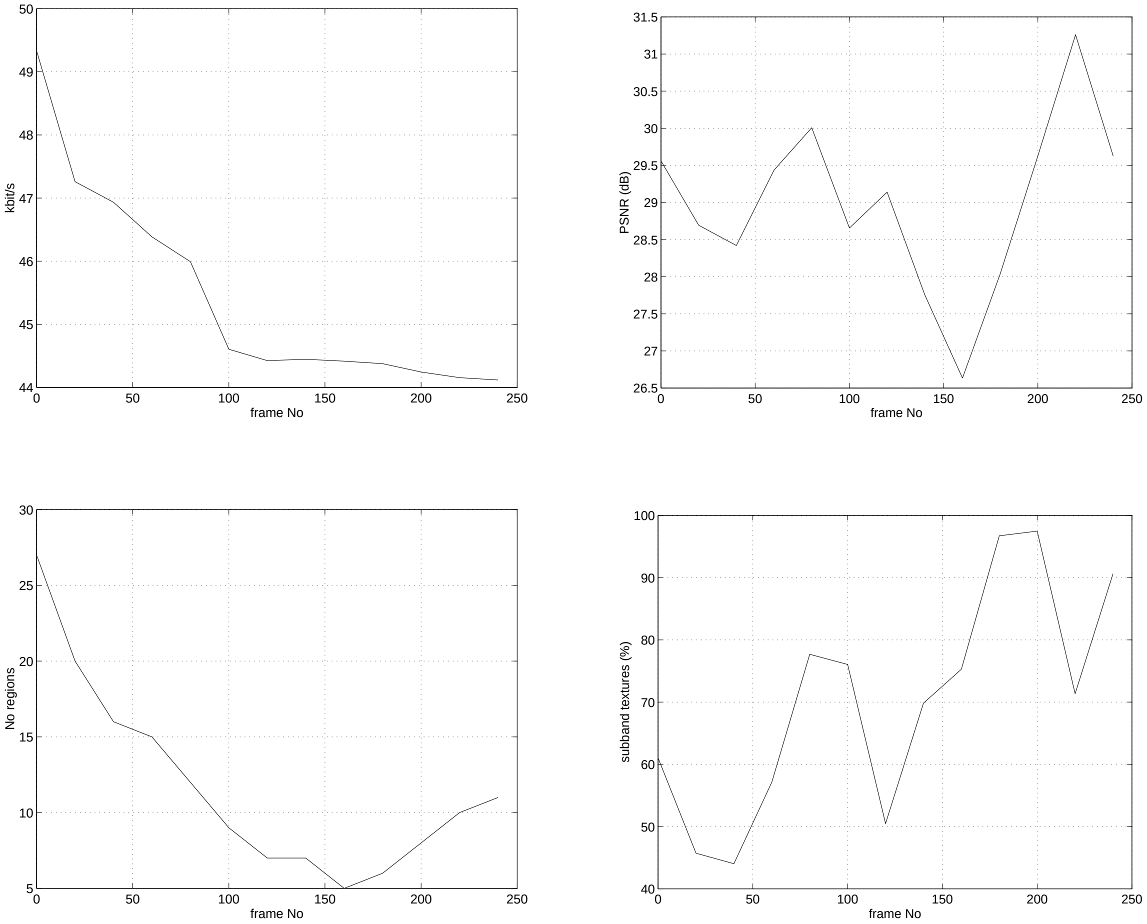


Figure 6.16: Evolution of rate, PSNR, No. of regions and portion of coded textures for the *foreman* sequence

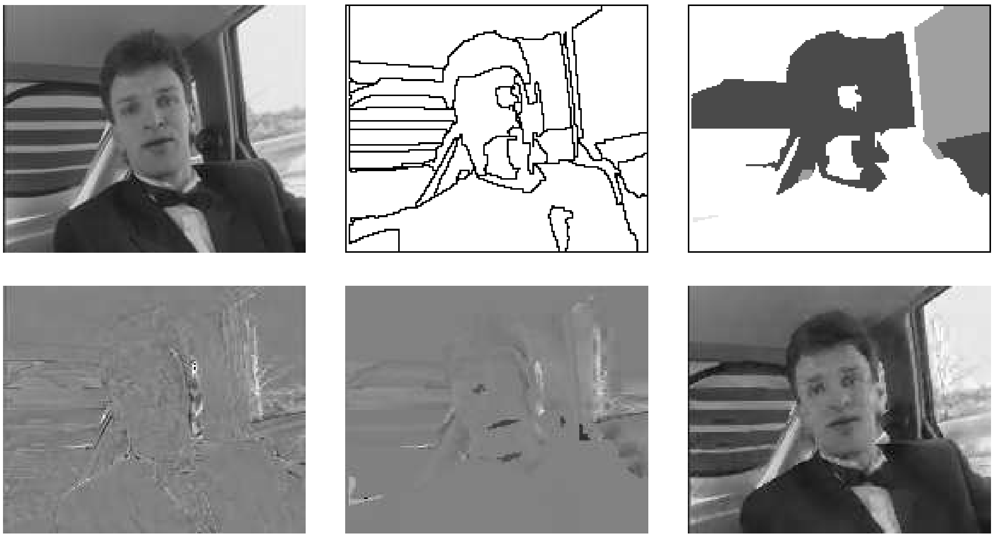


Figure 6.17: Results on video sequences: *car-phone* (frame No. 48). First row: original, partition and decision map. Second row: prediction error in inter-frame regions, coded prediction error (and details) and coded frame (30 kbit/s)

The time evolution of the bit-rate, PSNR values, number of regions and bit-rate devoted to the coding of details are shown in the plots of Fig. 6.18 for the sequence *car-phone*. In this case, the bit-rate includes the rate for the details component. The behavior of the plotted figures may be explained as before. The *car-phone* sequence does not present a sudden camera motion as the *foreman* sequence. Therefore, except for the first frame coded with a higher number of bits (12000 bits), the PSNR values are kept rather constant between 30 and 32 dB (notice the scaling of the Y axis in the corresponding plot). It is worthwhile to notice the behavior of the modified READ coding for the coding of detail positions, that improves its performance along the time, as shown in the bottom right plot.

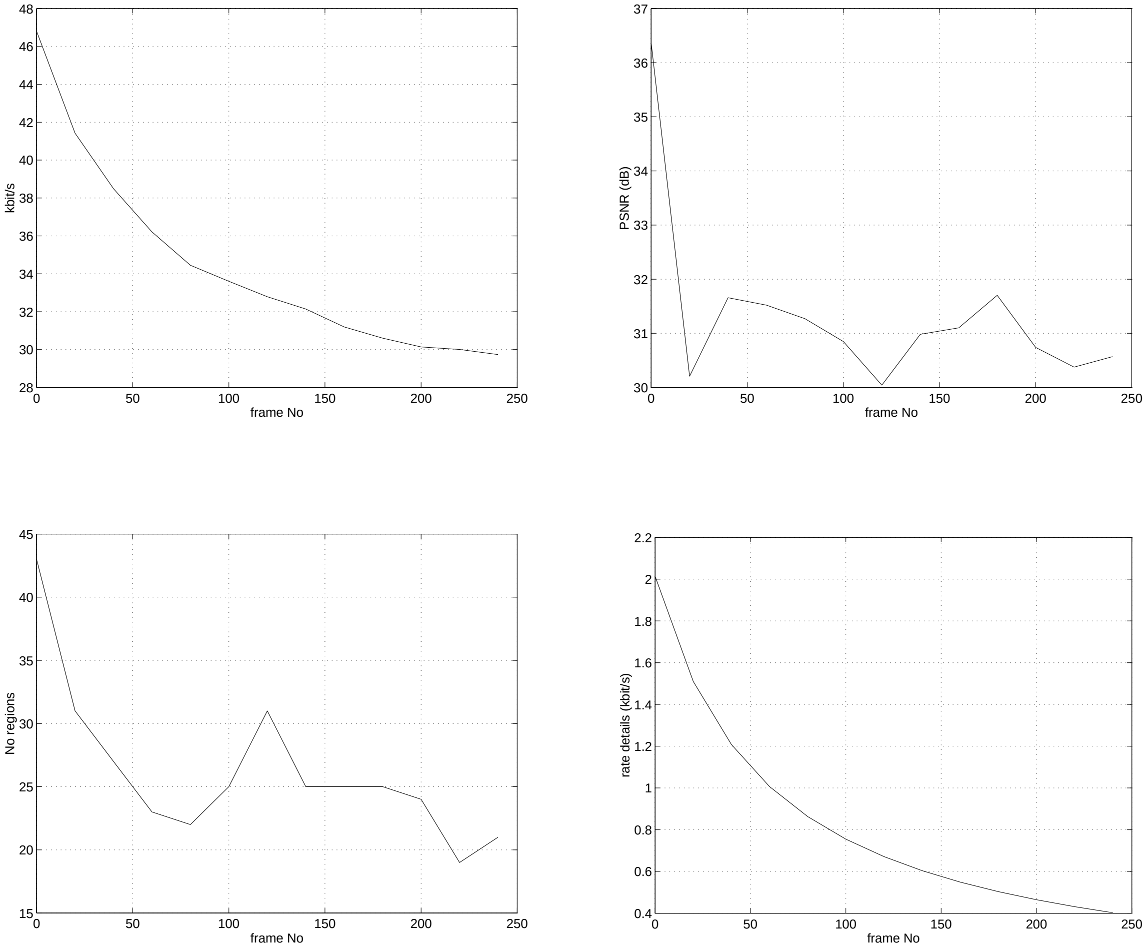


Figure 6.18: Evolution of rate, PSNR, No. of regions and details rate for the *car-phone* sequence

Chapter 7

Conclusions

The work of this thesis has been devoted to the study of perceptual image and video coding techniques. Perceptual image coding is based on the efficient representation of visual information using image models strongly related to the manner the Human Visual System perceives and understands visual information. The relevance of this research is due to the important role that images and video play in our civilization at the present time.

A 'perceptually motivated' model is put forward for coding applications. The first component of the model is formed by the strong edges and smooth areas of the image. The second component contains the small features or details that are meaningful for the visual system. Finally, the third component of the model is aimed at the representation of the fine textures of the image. We would like to highlight that we are aware that the proposed model is far for being a complete perceptual coding model for images and video sequences. The research effort has been directed to the development of the texture coding techniques that have been presented throughout this thesis. Each one of these techniques offers interesting possibilities in the framework of perception-based image compression schemes. Each one has proven to be efficient in the framework it was designed for. A great effort to fit them in a perceptual image model is still necessary in order to select whether the best match is the one that has been proposed or not. However, we hope that the reader will not be deceived for the lack of such optimized model. In our opinion, the results justify that the work which has been carried out so far has yield encouraging results.

Three texture coding techniques have been investigated. The first one is based on "strong edges" or significant spatial transitions in the image and the interpolation of smooth areas. The second one deals with the small image details. The third technique efficiently represents the fine textures of the image. Results both on still images and video sequences have been presented to assess the performance of these techniques. It has been shown that perceptual considerations have to be taken into account in the design of efficient image coding systems.

7.1 Summary of developments

The three main contributions of this thesis are listed below. We would like to stress the novelty of the texture coding techniques that have been investigated.

- Morphological interpolation
  A new interpolation method based on morphological operators has been proposed. From the point of view of computational efficiency, morphological interpolation is, at least, one order of magnitude faster that existing methods for scattered data interpolation, while yielding similar visual coding results. It has been applied to the coding of the primary component of the perceptual image model.

  A feature extraction technique based on the morphological watershed has been proposed for the extraction of the strong edges of the images.

- Region based subband coding
  The application of subband coding techniques to arbitrary shaped image regions has been proposed. A symmetric signal extension technique has been developed for the analysis and synthesis filter banks. The results assess the superior reconstruction quality of the region-based subband coding scheme with respect to different techniques that have been proposed for texture coding. Region-based subband analysis fully exploits the information about the edge structure of the images. This produces clear subjective improvements over conventional subband coding, which are especially noticeable at low and very-low bit-rates. Region-based subband analysis yields a twofold decomposition: in the spatial and in the frequency domains. Rate-constrained quantizer optimization may be performed on both domains independently, so that the coding of the region-based frequency decomposition can be made fully *adaptive* to the information contents of each region and, besides, to the subjective perception of such contents by the visual system.

- Perceptual extraction, ranking and selection of image details
  A new method for coding small meaningful image features in very low bit-rate image and video coding applications has been presented. Small visual features improve the subjective quality of the reconstructed images at a minimum cost. The features are extracted using morphological operators in the spatial-temporal domain. Once extracted, a perceptual selection is performed in order to keep only the most significant ones. An

efficient coding technique has been proposed for the coding of these features, which is based on motion compensation and relative addressing. The novelty of the scheme is the analysis of the perceptual significance of image features. The results prove that explicit perceptual measures of visual components should be considered in advanced second generation in order to reach the target compression for very low bit-rates video coders.

The integration of the above mentioned techniques in a perceptual model of the image has been studied. The application of the perceptual model to images and video sequences has yield encouraging results. For video sequences presenting moderate motion, such as *foreman* or *car-phone*, bit-rates of the order of 35 to 45 kbit/s have been obtained with good perceptual quality, and PSNR values about 30 dB. The segmentation-based coding scheme that has been employed allows the adaptation of the coding techniques to the image contents and the available bit-rate. As the bit-rate decreases, morphological interpolation (strong edge component) prevails over region-based subband coding. The perceived quality decreases gradually and annoying artifacts such as ringing, blurring or false contours appear smoothly. At higher bit-rates, the quality of the images improves significantly because of the efficiency of the region-based subband coding scheme increasing the rendition of the coded textures.

7.2 Current work and future research lines

Currently, the research effort is devoted to the validation of the perceptual three-component model of the image. As pointed out at the beginning of this chapter, heuristic reasons have led us to the choice of this perceptual model. However, extensive study is still necessary to validate the perceptual model. Regarding the particular coding techniques, further lines of research as the ones listed below can be addressed:

- The application of the morphological interpolation algorithm to other fields different from image coding. Actually, a formulation of the implemented algorithm in terms of non-linear Partial Differential Equations (PDE's) is in progress at the time of writing. Geometric PDE's became a major topic of research in the past years. The advantages of PDE's or curve/surface evolution approaches in image analysis is that they allow to think about image processing in the continuous domain. The problem in hand is then approached as an image deformation task. This may yield novel solutions to classical problems such as edge detection, filtering, anisotropic diffusion, and so on. The idea implemented in morphological interpolation in order to speed up linear diffusion processes may be translated to similar problems in this field.

- The study of morphological interpolation as a reconstruction filter for 'subband' coding. The initial steps in this direction have already been done. The advantage of morphological interpolation as a reconstruction operator is that it enables image interpolation from any structure of the sampled set. Therefore, the progressive down-sampling of the original image can be performed in any order, i. e. not only the conventional progressive decimation of 4:1. This allows a particular progressive reconstruction by interpolation of the original image starting from, for example, a single point and progressively adding new points taken from the reconstruction residues. This can be applied for the coding of region textures if an ordering of the region points is established a priory.

- The improvement of the detail extraction step including motion information in order to track the details along the time and not only using temporal connectivity.

- The study of different signal extension techniques for region-based subband coding. Higher order extensions than symmetric extension present higher computational load and it has been found [108] that for low bit-rate applications simple symmetric extension has good performance. However, further research from the point of view of computing the missing samples as has been implemented in this thesis for region-based subband coding is still necessary for higher order extensions.

Appendix A

Morphological operators

In order to address the problem of extracting significant features from digital images, analysis techniques strongly related to the physical image structure are required. Such techniques should deal with the 'shapes' contained in the image or in the video signal. *Mathematical morphology* provides processing tools that give a good insight into the structure of the images for feature extraction purposes.

Mathematical morphology is a non-linear processing technique originated from the work of Matheron and Serra [103], [104]. The morphological theory has sound mathematical foundations, but it can be used successfully with a very intuitive approach. Its original aim was to characterize physical properties by means of visual information. Although many signals combine additively, visual signals obey a very different way of composition. The physical world around us is generally made of opaque objects. Objects in the scened are 'ordered' in the sense that the nearest object is located before than the one located behind it. Therefore, the first prerequisite that a visual-like transformation must fulfill is to preserve the existing ordering relations between every pair of objects; i.e., it must be increasing instead of linear. The two notions are incompatible.

A review of some morphological operators that are used through this thesis is given in this appendix. The specific use of the watershed algorithm for the extraction of strong edges in order to perform the coding of the strong component of still images by means of open contour features is explained along with the applications of the morphological interpolation technique in chapter 3, and the morphological operator for the extraction of image details is explained in the corresponding section of chapter 4

A.1 Basic definitions

In mathematical morphology, the useful operations are those preserving the ordering relation defined in the working structure, a complete *lattice*, and commuting with the fundamental *laws*, the 'sup' and the 'inf'. These operations are said to be *increasing* transformations. Let us briefly state these concepts in the following.

A *complete lattice* is a set $\mathcal{P}$ such that,

1. a partial ordering relation $\leq$ is defined:

$$\begin{array}{lcl} A \leq A & & \\ A \leq B, B \leq A & \Rightarrow & A = B \\ A \leq B, B \leq C & \Rightarrow & A \leq B \end{array} \tag{A.1}$$

2. for each family of elements $\{X_i\} \in \mathcal{P}$, a 'sup' and and 'inf' exist:

$$\text{inf} \quad : \quad \text{maximum lower bound} \quad \wedge \{X_i\} \tag{A.2}$$

$$\text{sup} \quad : \quad \text{minimum upper bound} \quad \vee \{X_i\} \tag{A.3}$$

The increasing property states that if an ordering relation holds for two elements X_i and X_j which are input to the transformation ψ, the same ordering is kept for the output, i.e.,

$$\text{if} \quad X_i < X_j \quad \Rightarrow \quad \psi(X_i) < \psi(X_j)\,. \tag{A.4}$$

The basic operations preserving the ordering relation are *dilation* and *erosion*:

$$\text{Dilations} \quad \delta(\vee\{X_i\}) = \vee\{\delta(X_i)\} \quad \text{commute with the sup} \tag{A.5}$$

$$\text{Erosions} \quad \varepsilon(\wedge\{X_i\}) = \wedge\{\varepsilon(X_i)\} \quad \text{commute with the inf.} \tag{A.6}$$

Examples of lattices are the lattice of subsets $\mathcal{P}(E)$ of a set E, with the partial ordering defined by the inclusion law, or the lattices of real numbers or integer numbers, where the order is the natural order between levels (total ordering).

Originally, mathematical morphology was defined on sets, and later was extended to deal with numerical functions. The link between sets and functions is established by defining a function as a stack of decreasing sets. Each set is the intersection between the function and a plane of constant level:

$$X_f(\lambda) = \{x \in \mathcal{R}, \quad f(x) \geq \lambda\} \qquad \Leftrightarrow \qquad f(x) = Sup\,\{\lambda \quad \text{such that} \quad x \in X_f(\lambda)\} \tag{A.7}$$

Then, if f is a function, the following inclusion holds:

$$\forall \mu \leq \lambda \in \mathcal{R}, \quad X_f(\lambda) \subset X_f(\mu) \tag{A.8}$$

Therefore, any increasing transformation applied level by level to the family (stack) of sets parameterized by λ will result in another stack of sets (another function) preserving the ordering defined in the above equation A.8. Taking into account this equivalence between sets and functions, 'grey level' morphological operators have been defined to deal with functions and, in particular, with any signal x_i defined on the points i of an N-dimensional space such as a grey level image.

For comparison, in linear signal processing, the useful operations are also those preserving the working structure, in this case the *vectorial space*, and commuting with the fundamental laws, the addition and the scalar product. The resulting operation is linear instead of increasing: the *convolution*.

$$\psi\left(\sum_i a_i X_i\right) = \sum_i a_i \psi\left(X_i\right). \tag{A.9}$$

Linear filtering techniques modify the object intensities and, therefore, the estimated location of their corresponding contours. Morphological filters examine the geometrical structure of images by probing their micro-structure with certain elementary form, the *structuring element*, in the manner in which it fits into the image structure. Thus, the analysis is geometric in nature, and derives quantitative measures from this point of view.

Some basic definitions that are used through the chapters of the this thesis report are reviewed in the current appendix. The reader is referred to [103] and [104] for a complete explanation of the concepts of mathematical morphology.

A.2 Morphological filters

Morphological filters are defined as *increasing* and *idempotent* transformations. Being increasing, they preserve the ordering relation in the working space. The idempotence property limits the information loss by transforming in a single pass any original signal into an invariant signal. Morphological opening and closing are examples of morphological filters. They are based on the operations of *dilation* and *erosion*, which are defined below.

If x_i and yi denote two signals defined in an N-dimensional space, E^N, the erosion ε_n and dilation δ_n of x_i by a window or flat structuring element B of size n are given by:

- erosion

$$y_i = \varepsilon_n (x_i) = \inf_{k \in B} [x_{i+k}] \tag{A.10}$$

- dilation

$$y_i = \delta_n (x_i) = \sup_{k \in B} [x_{i+k}] \tag{A.11}$$

where i indicates the location of the current sample and k defines the distance to adjacent samples within the window. From a practical point of view, all the morphological filters in this appendix use a square structuring element whose square size is represented by n.

Morphological opening γ_n and closing φ_n of size n are based on dilation and erosion definitions:

- morphological opening

$$y_i = \gamma_n (x_i) = \delta_n (\varepsilon_n (x_i)) \tag{A.12}$$

- morphological closing

$$y_i = \varphi_n (x_i) = \varepsilon_n (\delta_n (x_i)) \tag{A.13}$$

Opening and closing are dual filters in the sense that the result of the closing is also the complement of the result of the opening applied to the complement of the signal. The opening (resp. closing) simplifies the original signal by removing the small bright (resp. dark) components where the structuring element does not fit. In addition, the contours of the large image components are often modified to fit the shape of the structuring element. In order to allow a perfect preservation of the contour information, a *reconstruction process* must to be used [104]. Its goal is to restore the contour of the objects that have not been totally eliminated by the opening or the closing. Let us describe this reconstruction process.

Two dual reconstruction processes may be defined. After an opening, a positive reconstruction is defined based on geodesic dilation, whereas in the case of closing, it is a negative reconstruction relying on geodesic erosion. Both geodesic operators need an input signal x_i and a reference signal r_i. Their definitions are given for unitary size (the smallest window size in digital case) usually taken as a 3×3-window in image processing:

- geodesic dilation of size 1

$$y_i = \delta^{(1)} (x_i, r_i) = \inf [\delta_1 (x_i), r_i] \tag{A.14}$$

- geodesic erosion of size 1

$$y_i = \varepsilon^{(1)} (x_i, r_i) = \sup [\varepsilon_1 (x_i), r_i] \tag{A.15}$$

These elementary geodesic dilation and erosion operators allow the introduction of the reconstruction processes, which are defined as iterated geodesic dilations or erosions. In practice, the unitary operations are iterated until idempotence, that is, until no change is observed in the output signal. Practical implementation of these operators may be done by means of efficient algorithms based on list structures that avoid any iterating process and lead to extremely fast algorithms [120].

- positive reconstruction

$$y_i = \gamma^{(rec)}\left(x_i, r_i\right) = \delta^{(1)}\left(\delta^{(1)}\left(\ldots\delta^{(1)}\left(x_i, r_i\right)\ldots, r_i\right), ri\right) \tag{A.16}$$

- negative reconstruction

$$y_i = \varphi^{(rec)}\left(x_i, r_i\right) = \varepsilon^{(1)}\left(\varepsilon^{(1)}\left(\ldots\varepsilon^{(1)}\left(x_i, r_i\right)\ldots, r_i\right), ri\right) \tag{A.17}$$

In morphological image processing, the function to be rebuilt by the reconstruction process is a ‘marker’ image for significant components of the reference image, whose locations are known but their exact shapes are not. Markers are, indeed, binary images identifying the presence of desired components. The original contours of these components will be found by means of the reconstruction process applied to the marker image, taking the original image as the reference function. Finally, the opening by reconstruction $\gamma^{(rec)}$ and closing by reconstruction $\varphi^{(rec)}$ are given by:

- opening by reconstruction

$$y_i = \gamma^{(rec)}\left(\varepsilon_n\left(x_i\right), x_i\right) \tag{A.18}$$

- closing by reconstruction

$$y_i = \varphi^{(rec)}\left(\delta_n\left(x_i\right), x_i\right) \tag{A.19}$$

Their simplification effects in the filtered images are smaller than those of the morphological opening and closing. Large bright (resp. dark) objects which have not been completely eliminated by the morphological opening (resp. closing) are rebuilt to their original shape by the geodesic process so that their contours are preserved.

A.3 The morphological Laplacian

The morphological Laplacian, $L(f)$, is defined as the residue of the gradient by dilation, $g^{+}()$, and the gradient by erosion, $g^{-}()$:

$$g^+(x_i) = \delta(x_i) - x_i \tag{A.20}$$

$$g^-(x_i) = x_i - \varepsilon(x_i) \tag{A.21}$$

$$L(x_i) = g^+(x_i) - g^-(x_i) \tag{A.22}$$

The morphological Laplacian is greater than zero at the lower edge of the transitions and smaller than zero at the upper edge. In flat surfaces or slanted planes without convexity changes, it cancels out. Indeed, it can be shown that the morphological Laplacian is an approximation of the signal second derivative.

Appendix B

Quadrature Mirror Filters

The two-channel quadrature mirror filter (QMF) bank, shown in Fig. B.1, is one of the earliest and most commonly employed structures for subband coding [23]. The analysis bank is composed of two frequency selective filters: a low-pass filter $H_0(z)$ and a high-pass filter $H_1(z)$, which split the incoming sequence $x(n)$ into a low-pass signal $x_0(n)$ and a high-pass signal $x_1(n)$. The name quadrature mirror filter derives from the fact that the response of $H_1(z)$ is the *mirror-image* of the response of $H_0(z)$ with respect to frequency $\pi/2$, which is a *quarter* of the sampling frequency.

To preserve the system sampling rate, both channels are critically decimated by factors of two. The decimated signals are typically encoded and transmitted. At the receiver end, the signals are decoded and passed through the interpolators. The decimator-interpolator cascade causes aliasing and imaging. The purpose of the analysis filters $H_0(z)$ and $H_1(z)$ is to avoid the aliasing effect due to decimation (down-sampling), whereas the synthesis filters $F_0(z)$ and $F_1(z)$ eliminate the 'images' caused by interpolation (up-sampling). As a result, the signals $v_0(n)$ and $v_1(n)$ are good approximations of $x_0(n)$ and $x_1(n)$ and the reconstruction $\hat{X}(z)$ resembles $X(z)$ closely.

In order to avoid aliasing, the responses of $H_0(z)$ and $H_1(z)$ must be disjoint. On the other hand, to ensure the reconstruction of the input signal, the the set of filters of the analysis bank should cover the whole frequency range. The only obvious solution is to make the responses very sharp, approximating the ideal response, but it is well-known that sharp cut-off filters require very high order, are highly sensitive to quantization and often cause instability problems (if IIR).

The classic QMF solution adopted in order to overcome this problem is to permit aliasing

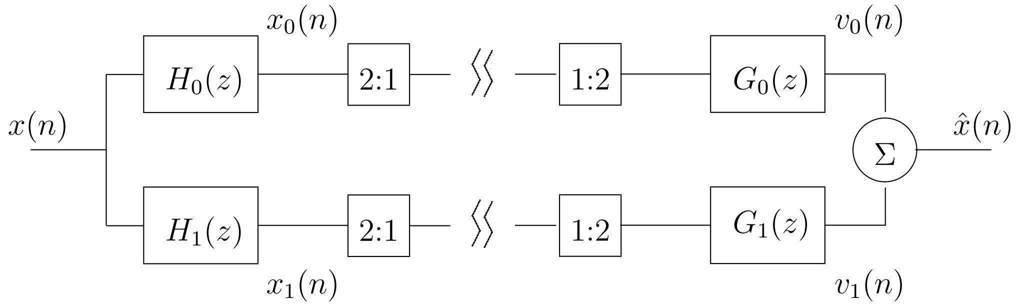


Figure B.1: Two-channel Quadrature Mirror Filter bank

at the output of the decimators by designing the analysis filters with their responses overlapping slightly around $\pi/2$. Then, the synthesis filters $F_0(z)$ and $F_1(z)$ are chosen such that the imaging produced by the interpolators cancels the aliasing effect. In fact, *exact cancellation* is possible. In the following, the filter design equations for the QMF filter bank are reviewed. Some proposed solutions for subband frequency splitting are also presented and compared. The last section discusses the extension of one-dimensional QMF filtering techniques for image signals.

B.1 Analysis of the 1-D QMF filter bank

The QMF filter bank is a *multi-rate* digital filter bank, where decimators and interpolators change the sampling rate throughout the system. The input-output relation for a two-fold decimator can be written in the z-transform domain as follows [63, ch. 2, section 6]:

$$Y(z) = \frac{1}{2}\left[X(z^{1/2}) + X(-z^{1/2})\right] \tag{B.1}$$

The 'aliasing' effect is due to the second term in eq. B.1, which is a shifted version (or alias) of the first term by an amount of 2π in the frequency domain. On the other hand, a two-fold interpolator causes shrinking in the frequency axis, known as 'imaging' effect, that can be expressed with the following simple equation:

$$Y(z) = X(z^2) \tag{B.2}$$

In the case of the two-band filter bank shown in Fig. B.1, and based on the relations B.1

and B.2, the system equations can be written as follows:

$$\hat{X}(z) = T(z)X(z) + A(z)X(-z) \tag{B.3}$$

with $T(z)$ being the distortion function

$$T(z) = \frac{1}{2}\left[H_0(z)G_0(z) + H_1(z)G_1(z)\right] \tag{B.4}$$

and $A(z)$, the aliasing term

$$A(z) = \frac{1}{2}\left[H_0(-z)G_0(z) + H_1(-z)G_1(z)\right] \tag{B.5}$$

Note that it is not possible to write down an expression for $\hat{X}(z)/X(z)$ that is independent of $X(z)$ itself. This is not surprising since the QMF bank is not invariant, as the decimators and interpolators are (time- or space-) variant systems [115].

B.1.1 Exact aliasing cancellation

The term $A(z)$ in B.5 represents the effects of aliasing and imaging. This term can be made to disappear simply by choosing the synthesis filters to be

$$G_0(z) = H_1(-z) \qquad G_1(z) = -H_0(-z) \tag{B.6}$$

Once the aliasing is cancelled, i.e. $A(z) = 0$, the QMF bank becomes a linear and invariant system with overall transfer function

$$\begin{aligned} \frac{\hat{X}(z)}{X(z)} = \quad T(z) \quad &= \frac{1}{2}\left[H_0(z)H_1(-z) - H_1(z)H_0(-z)\right] \\ &= \frac{1}{2}\left[F(z) - F(-z)\right] \end{aligned} \tag{B.7}$$

where $F(z)$ is defined as the *product filter*

$$F(z) = H_0(z)H_1(-z) \tag{B.8}$$

In order to obtain an efficient implementation (see section B.1.3 below) QMF's are defined to be related by a frequency shift of π, i.e,

$$H_1(z) = H_0(-z) \tag{B.9}$$

For this class of analysis/reconstruction filter pair, the product filter becomes $F(z) = H_0^2(z)$ or $F(-z) = H_1^2(z)$ and the transfer function

$$T(z) = \frac{1}{2}\left[H_0^2(z) - H_1^2(z)\right] = \frac{1}{2}\left[H_0^2(z) - H_0^2(-z)\right] \tag{B.10}$$

In summary, the choice of the filters according to

$$H_1(z) = H_0(-z) \qquad G_0(z) = H_0(z) \qquad G_1(z) = -H_0(-z) \tag{B.11}$$

leads to complete cancellation of aliasing.

B.1.2 Perfect reconstruction property

Ideally, the transfer function $T(z)$ should be a pure delay of the form $T(z) = z^{-\delta}$ so that the reconstructed signal is a delayed version of $x(n)$. In this case, the filter bank fulfills the *perfect reconstruction property.* Since $T(z)$ is in general not a delay, it is called distortion function. The quantity $\left|T(e^{j\omega})\right|$ is the amplitude distortion and $arg\left[T(e^{j\omega})\right]$ is the phase distortion.

Because of the importance of phase in images [60, pp. 31–39], and thus to avoid phase distortion in subband coding schemes, linear phase filters are often desirable. Clearly, if $H_0(z)$ and $H_1(z)$ are linear phase filters, then $T(z)$ given by B.10 has linear phase as well. Assuming $H_0(z)$ to be a linear phase low-pass FIR filter of order $N-1$, it can be expressed [63, ch. 4, section 5] as $H_0(e^{j\omega}) = e^{-j\omega(N-1)/2}H_{0,r}(e^{j\omega})$, where $H_{0,r}(e^{j\omega})$ is the (real-valued) amplitude response. Then, the frequency response $T(e^{j\omega})$ takes the form:

$$\begin{aligned} T(e^{j\omega}) &= \frac{e^{-j\omega(N-1)}}{2}\left[H_{0,r}^2(e^{j\omega}) - (-1)^{N-1}H_{0,r}^2(e^{j(\pi+\omega)})\right] \\ &= \frac{e^{-j\omega(N-1)}}{2}\left[\left|H_0(e^{j\omega})\right|^2 - (-1)^{N-1}\left|H_1(e^{j\omega})\right|^2\right] \end{aligned} \tag{B.12}$$

If the length N of the filter is odd (order N-1, even) then, at the frequency $\omega = \pi/2$, the response $T(e^{j\omega})$ given by B.12 is zero and this implies severe amplitude distortion. Accordingly, with the choice of filters as in B.11, the linear phase FIR filter $H_0(z)$ must be *even* in length. The residual amplitude distortion is then

$$\left|T(e^{j\omega})\right| = \frac{1}{2}\left[\left|H_0(e^{j\omega})\right|^2 + \left|H_1(e^{j\omega})\right|^2\right] \tag{B.13}$$

In the case of linear phase filters, exact reconstruction requires $\left|H_0(e^{j\omega})\right|^2 + \left|H_1(e^{j\omega})\right|^2$ to be constant for all ω. In general, for the filters defined in B.11, the perfect reconstruction condition is stated as follows:

$$H_0^2(e^{j\omega}) - H_1^2(e^{j\omega}) = 2 \tag{B.14}$$

Table B.1: Parameters of Johnston's 8 TAP(A) QMF filters

parameter	value
normalized transition band	0.14
pass-band ripple	0.06 dB
stop-band rejection	31 dB
phase characteristic	linear
perfect reconstruction	nearly
regularity (zeros at $z = -1$)	none

There exist two simple cases where this condition is satisfied. The first is the case where the analysis filters are infinitely long, ideal half-band filters. Obviously, the resulting analysis/synthesis system is distortion-less but not very useful for real implementations. The second case occurs when $H_0(e^{j\omega})$ and $H_1(e^{j\omega})$ are of order one of less. For example,

$$H_0(z) = \frac{1}{\sqrt{2}}(1 + z^{-1}) \qquad H_1(z) = \frac{1}{\sqrt{2}}(1 - z^{-1}) \tag{B.15}$$

Although these filters form a perfect reconstruction filter bank, they lack the frequency resolving power of higher order filters. Higher order QMF's will never permit reconstruction to be exact, although the distortion can be made to be very small.

B.1.3 Johnston filters

The set of QMF filters designed by Johnston [48] have been widely used in the image coding community, partially because of the simplicity of the design technique and the published tables of filter coefficients. With the restrictions expressed in B.11, Johnston designed a set of filter banks based on the minimization of a weighted sum of the reconstruction error and the stop-band energy. Johnston QMF's are FIR, even in length and linear phase. They perform exact aliasing cancellation and have good reconstruction properties. The frequency responses of the 8 TAP low-pass and high-pass Johnston filters are shown in Fig. B.2. Notice the 3 dB point at a quarter of the sampling frequency. Some features of the 8 TAP (A) filters designed by Johnston are given in the table B.1.

When applied in a two-band system, Johnston filters result in an overall system response with only a negligible degree of distortion. The amplitude distortion $|T(e^{j\omega})|$ of the 8 TAP QMF filter bank is shown in Fig. B.3. As this magnitude is not constant for all ω, the filters

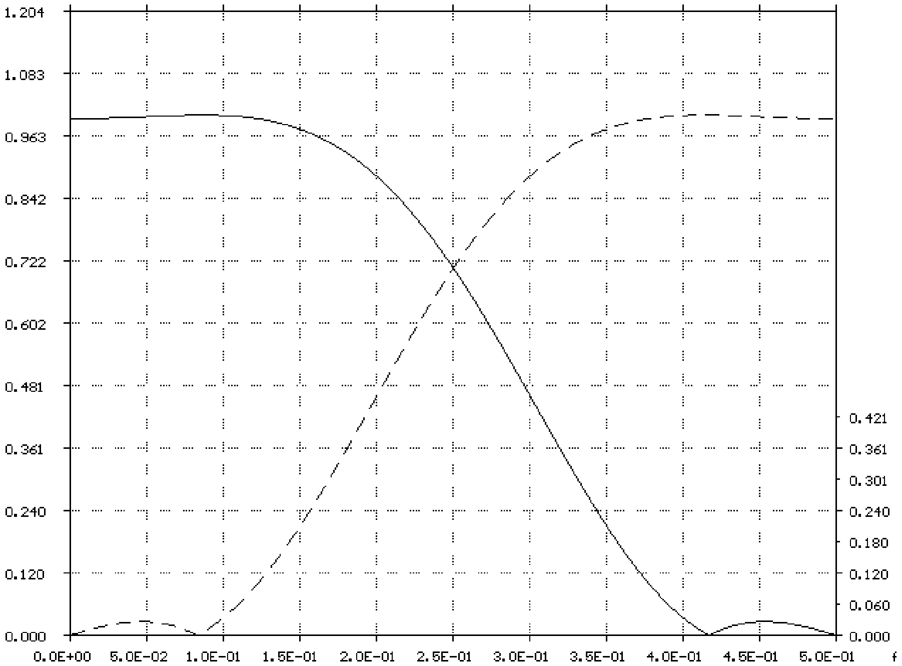


Figure B.2: Frequency responses of the 8 TAP Johnston QMF filters

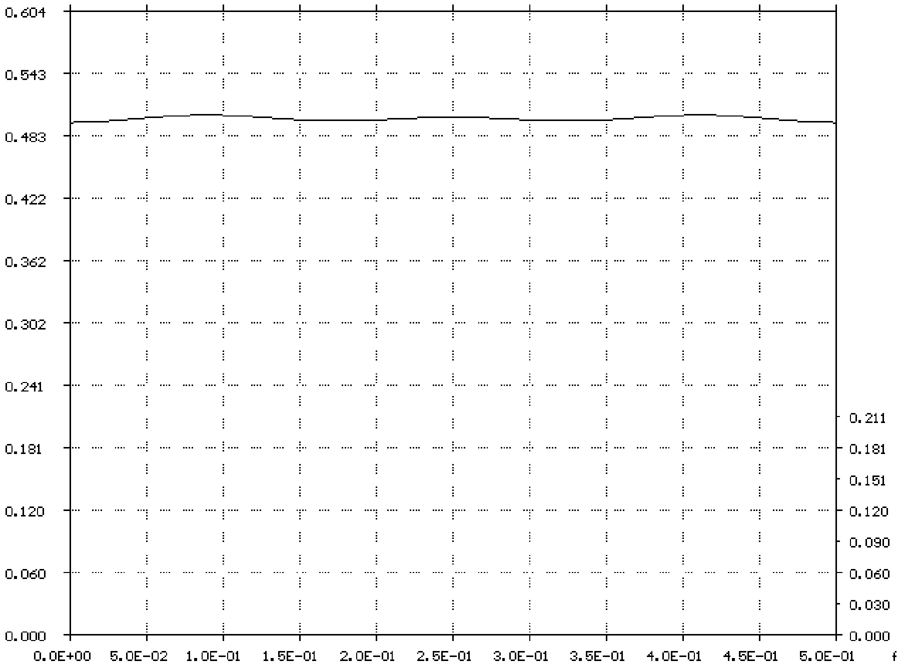


Figure B.3: Overall amplitude response of the 8 TAP Johnston QMF filter bank

are not perfect reconstruction filters, but they approximate the target expressed in B.14. Johnston's QMF solution has the advantage that it can be implemented in an efficient way because the coefficients of high-pass and low-pass filters are identical in magnitude but may have a different sign:

$$h_1(n) = (-1)^n h_0(n) \tag{B.16}$$

This coefficient property is derived from the fact that QMF filter are frequency shifted versions of each other (B.9), which may be formally exploited by using the implementation known as polyphase structure [119]. The polyphase implementation has the effect of reducing the number of multiplications by 50 percent.

B.2 Other solutions for the filter bank

Besides QMF filters, a number of different filtering techniques have been proposed for the analysis/synthesis filter banks of the classical subband coding scheme. Some of them are reviewed in this section and compared against the classical QMF solution.

- *Conjugate Quadrature Filters*
  There exist a class of FIR filters [109], called *conjugate quadrature filters* (CQF's), that preserves the perfect reconstruction property of B.14 and also allows for the design of realizable filter banks with arbitrary frequency resolution. These filters are designed by removing the constraint in B.9 whereas the alias cancellation condition stated in B.6 is kept. The design technique consists of directly designing the product filter $F(z)$ of B.8, and then factoring the result into separate analysis/synthesis filter banks (e.g., minimum and maximum phase components). CQF filters are constrained to have the same magnitude response, which is exactly the square root of $F_0(e^{j\omega})$, so that the product filter is designed to be the square of the desired analysis filter magnitude.

  Conjugate quadrature filter analysis/reconstruction is distortion free, and CQF's have higher filter quality than QMF filters. In fact, they may be designed to be optimal filters with equi-ripple characteristic in the frequency intervals of interest according to the minimax or Chebishev criterion [74]. However, CQF's have non-linear phase and, since they are not frequency shifted versions of each other, these filters cannot be realized using the polyphase structure. In addition, in presence of quantization, CQF's present more Gibbs effects (ringing) than QMF's [6]. This is explained by the the larger oscillations or *ripples* of the filter response near the Nyquist frequency.

- *IIR Filters*
  The superior magnitude characteristics associated with IIR filters can be exploited in subband filter banks [108]. The main advantage of IIR systems is the lower computational complexity. IIR filters can achieve dramatic computational efficiency gains over FIR systems (beyond 10 to 1) for comparable performances. Although IIR filter banks are often sensitive to finite word length effects and numerical accumulation errors, they can be designed to have low numerical sensitivity.

  The main arguments against using IIR filters for subband coding of images is that they have non-linear phase and they show greater ringing distortion at low bit-rates [108]. As discussed below (see section B.2.1), the ringing distortion is directly related to the amplitude of the ripples in the step response of the low-pass filters, which produce pronounced overshoots followed by decaying ripples in the low-pass filtered signal transitions. By contrast, the step response of QMF's tend to have ripples with somewhat smaller amplitude and distributed on both sides of the step transition.

- *'Wavelet' Filters*
  CQF's and IIR filters were initially developed for speech processing and, according to the needs of this field, they are very selective but not 'smooth' in the frequency domain. The main difference of wavelet filters with traditional subband coding is that wavelet filters are chosen to be *regular* [89]. 'Regularity' can be interpreted as a *flatness*

condition of the frequency response at half the sampling frequency. The regularity criterion brought by wavelet theory over filter banks has been found to be relevant for image coding applications [88]. Daubechies' orthonormal wavelets bases filters are deduced from 'maximally flat' low-pass filters, with maximum number of zeroes at $z = -1$. Filters designed this way are CQF with high regularity (smoothness) and low selectivity. Actually, Daubechies' filters are not selective at all, but they present high attenuation with no oscillation and their phase behavior is the closest to linear for perfect reconstruction filters. In order to increase selectivity while keeping smoothness in the stop-band, Rioul [88] proposed a design algorithm for filters between Daubechies' and Smith/Barnwell CQF's in terms of regularity and selectivity.

- *Asymmetrical Filter Banks*
QMF's and CQF's constrain the filters $H_0(z)$ and $H_1(z)$ to have the same magnitude response in order to obtain a symmetrical decomposition of the frequency spectrum. On the contrary, the wavelet decomposition is obtained using a highly asymmetrical tree, although it yields a good frequency partition. Based on the observation that high-frequency information of natural images are edges, which have a very limited spatial support, whereas the low-frequency information comes from homogeneous regions that have a large spatial extension, Egger and Li [27] have recently proposed subband coding of images using *asymmetrical filter banks* (AFB). AFB's consist of a very short high-pass analysis filter and a longer low-pass counterpart. These filters are linear phase, maximally regular and of unequal length. AFB's present good step response of the longer low-pass filters, whereas the length of the proposed high-pass filters is very short, producing very few oscillations of the filtered image around the edges. This results in a reduced entropy of the high-pass subbands and makes the quantization error very localized around the edges and diminishes the ringing effects. Egger and Li have experimentally verified that AFB's give better visual quality than classical QMF filter banks, although their frequency selectivity is small.

B.2.1 Discussion

The performance of linear filters for subband image coding is characterized by the presence of *ringing* effects. The source of this distortion can be seen as the impact of the filters in terms of their step response. At low bit-rates, when quantization is carried out at the output of the analysis stage, the fine structure in the high frequency bands is often lost or, at best, severely distorted. Thus, a step edge being received by the synthesis filter in the low-pass channel produces the step response of the low-pass filter at the output. The low-pass step response inherently contains ripples, which are associated with the Gibbs phenomenon. At higher bit-rates, the fine grain structure in the high-pass channel would also produce ripples, which tend to cancel the ripples in the low-pass synthesis channel resulting in a smooth image edge. Therefore, the ringing distortion is directly related to the step response of the synthesis filter. If the amplitude of the ripples in the synthesis filter step response is reduced, the ringing will be reduced as well.

The design of filters with optimal magnitude response characteristics, as CQF's, implies the necessity of step response ripples. In the light of the alternation principle [74, p. 468], it is impossible to achieve small step response ripples and good low-pass frequency characteristics in a single filter. Some balance between frequency domain and step response characteristics may be achieved, but subband coding with linear filters will always present ringing distortion to some extent.

In addition, symmetrical perfect reconstruction filter banks (CQF's) do not present linear phase, what is true for IIR filters as well. Non-linear phase filters produce distortion that often takes the form of pixel intensity oscillations (ringing) in the vicinity of image edges [108, p. 130]. In general, subjective preference is given to FIR filters which have linear phase.

The wavelet approach is found at the other extreme of the design of optimal filter magnitude responses. Wavelet filters are designed to be smooth (regular) in the frequency domain. The regularity effects take place in the stop-band, where they present greater attenuation than QMF's and CQF's. However, some local artifacts arise along the edges due to the regularity constraint. More distortion in form of blurring is is introduced due to the lower cut-off frequency of wavelet filters.

Some authors [57] have reported the advantages of using short kernel filter banks in order to minimize ringing effects. Although the amplitude distortion of non-perfect reconstruction QMF filter banks can be made very small with the use of long kernels, long filters may increase the computational complexity, while not providing significant coding gain. The use of long kernel filters for video and image applications is discouraged [57]. These type of filters may be suitable for a small number of bands. However, their application to high compression image coding, which necessitates the decomposition in a large number of narrow bands, may affect the coding efficiency due to their poor frequency responses. In the case of high compression, excessive coding noise can have a greater effect on the cancellation of aliasing due to the large overlap between the neighboring bands.

An interesting comparison of filter banks for subband coding of images has been reported in [6]. The study is limited to eight tap FIR filters, which allow to limit computation time and lead to reasonable selectivity. The authors conclude that in a subband coding scheme the main filter families studied give the same kind of performance and that the influence of the phase characteristic, the regularity and the selectivity are not significant enough to justify an adaptation of the coding process to the analysis/decomposition filters. Sixteen tap filters produced similar results. The effective differences observed were mainly due to the phase characteristic. Eight tap QMF filters were reported to give good results when the high frequency sub-images are cancelled, what often occurs in low bit-rate applications.

Finally, the recently proposed asymmetrical filter banks show only small empirical improvement regarding the ringing effects and cannot be implemented as efficiently as QMF's. Actually, the same authors have proposed an adaptive subband decomposition scheme [28] that, in order to get rid of the annoying ringing distortion, switches from AFB's to morphological filter banks when non-textured regions or strong edges are encountered. Such adaptive scheme benefits from the superior performance of linear filters in textured areas.

B.3 Extension to two-dimensional signals

For two-dimensional signals, solutions based on separable filtering are the most computationally efficient and generally the most attractive. Separability enables to treat 2-D systems in terms of 1-D filtering concepts. 'Separability' is used here in a general sense and refers to any 2-D filter which can be implemented in terms of series of 1-D filtering operations performed along linearly independent image lines. One popular and straightforward implementation of 2-D separable filtering consists of first applying 1-D filtering and decimation to the rows of the image and then applying the same procedure to the columns of the resulting images. This results in a basic four-band decomposition as shown in Fig. B.4.

In addition to the computational benefits of separable filtering, all the desirable 1-D reconstruction properties carry over directly. This is clearly true since the rows and columns are sequentially reconstructed with 1-D filter banks. Similar properties have been derived for filter banks that are separable in a less restrictive sense than row/column separability. The general framework of multi-dimensional subband filter banks (separable and not separable) has been treated by Vetterli [118].

Similarly to the one-dimensional case, a two-dimensional signal $x(m,n)$ sub-sampled by 2 along each axis results in a signal $y(m,n)$ that presents shifted versions of the spectrum $X(z_1,z_2)$ in the frequency domain (aliasing). The input-output relation of the 4:1 2-D orthogonal decimator can be written as follows:

$$Y(z_1,z_2) = \frac{1}{4}\left[X(z_1^{1/2},z_2^{1/2}) + X(z_1^{1/2},-z_2^{1/2}) + X(-z_1^{1/2},z_2^{1/2}) + X(-z_1^{1/2},-z_2^{1/2})\right] \tag{B.17}$$

The relation for the reciprocal 4:1 interpolator is expressed with the following z-transform:

$$Y(z_1,z_2) = X(z_1^2,z_2^2) \tag{B.18}$$

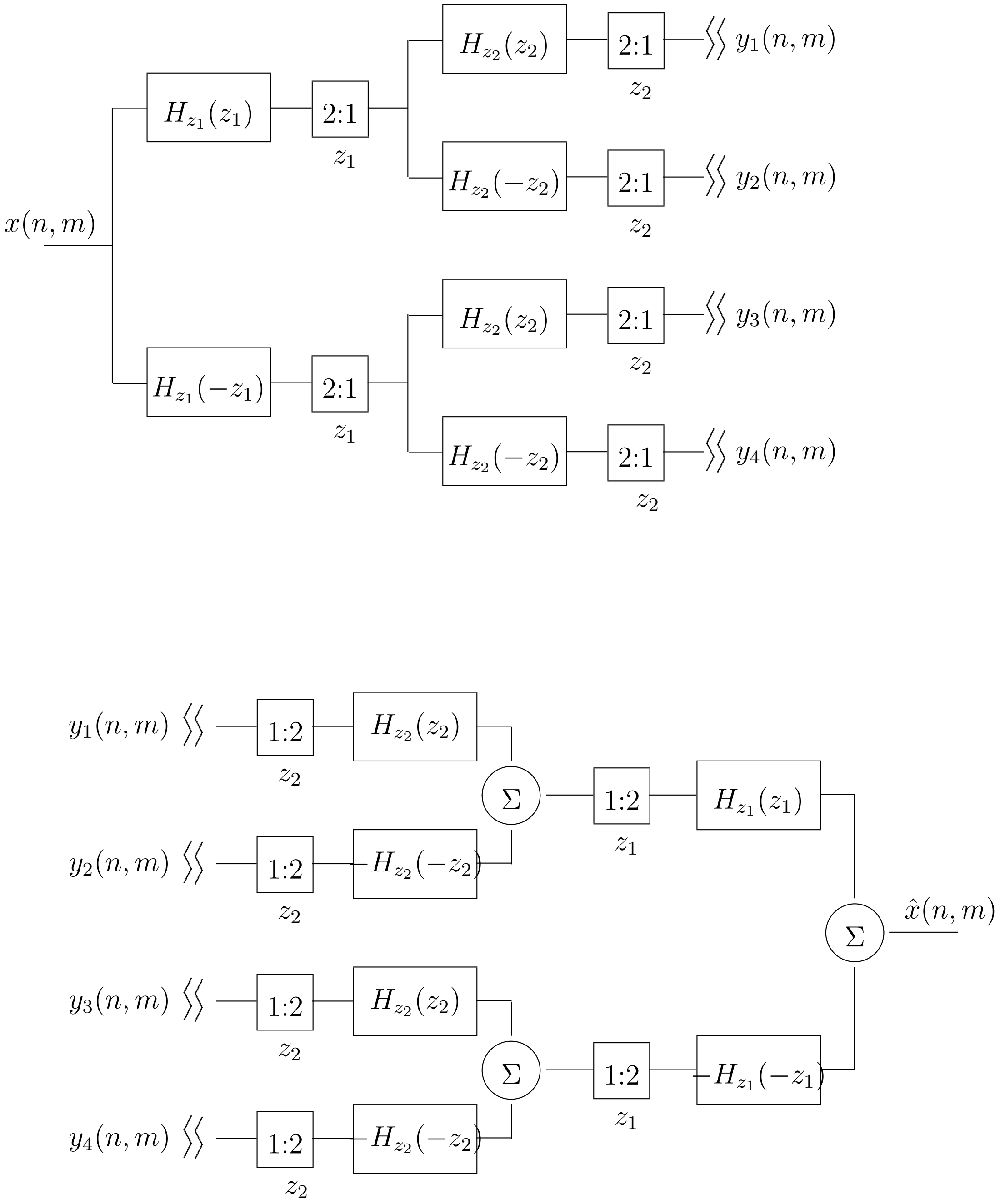


Figure B.4: Subband analysis/synthesis with separable filters

In the case of the four-band 2-D filter bank shown in Fig. 5.1 of chapter5 (page 88) and with the symmetry constraints stated in eqs. 5.1–5.4, the system equations are [118]:

$$\hat{X}(z_1, z_2) = T(z_1, z_2)X(z_1, z_2) + A(z_1, z_2)X(-z_1, -z_2) \tag{B.19}$$

with $T(z_1, z_2)$ being the distortion function

$$T(z_1, z_2) = \frac{1}{4}\left[H^2(z_1, z_2) - H^2(z_1, -z_2) - H^2(-z_1, z_2) + H^2(-z_1, -z_2)\right] \tag{B.20}$$

and $A(z_1, z_2)$ the aliasing term

$$A(z_1, z_2) = \frac{1}{2}\left[H(z_1, z_2)H(-z_1, -z_2) - H(z_1, -z_2)H(-z_1, z_2)\right] \tag{B.21}$$

A necessary and sufficient condition for the cancellation of the aliased components $A(z_1, z_2)$ in B.21 is the separability of the filter stated in eq. 5.5. Let us rewrite this equation below along with its expression for the impulse response of filter $h(n_1, n_2)$:

$$H(z_1, z_2) = H_{z_1}(z_1)H_{z_2}(z_2) \qquad h(n_1, n_2) = h_{n_1}(n_1)h_{n_2}(n_2) \tag{B.22}$$

In that case, the aliasing is cancelled and the system transfer function results:

$$\begin{aligned} T(z_1, z_2) \quad = \quad & \frac{1}{4}\left[H_{z_1}(z_1)H_{z_2}(z_2) - H_{z_1}(-z_1)H_{z_2}(z_2)\right. \\ & \left.-H_{z_1}(z_1)H_{z_2}(-z_2) + H_{z_1}(-z_1)H_{z_2}(-z_2)\right] \end{aligned} \tag{B.23}$$

In order to obtain perfect reconstruction, assuming that both filters $h_{n_1}(n_1)$ and $h_{n_2}(n_2)$ are even length and linear phase, the following condition should be met:

$$\begin{aligned} & H_{z_1}(e^{j\omega_1})H_{z_2}(e^{j\omega_2}) \\ & -H_{z_1}(e^{j(\omega_1+\pi)})H_{z_2}(e^{j\omega_2}) \\ & -H_{z_1}(e^{j\omega_1})H_{z_2}(e^{j(\omega_2+\pi)}) \\ & +H_{z_1}(e^{j(\omega_1+\pi)})H_{z_2}(e^{j(\omega_2+\pi)}) = 4 \end{aligned} \tag{B.24}$$

If sum of the squared magnitudes of the frequency responses is equal to a constant, the signal is perfectly reconstructed. Since the filter is separable, the sum of the one-dimensional filters has to be equal to a constant and thus conventional 1-D QMF's can be used, i.e.,

$$H^2_{z_1}(e^{j\omega_1}) - H^2_{z_1}(e^{j(\omega_1+\pi)}) = H^2_{z_2}(e^{j\omega_2}) - H^2_{z_2}(e^{j(\omega_2+\pi)}) = 2 \tag{B.25}$$

Appendix C

Segmentation-based rate-constrained optimization

This appendix briefly describes the segmentation scheme for the coding of images and video sequences that has been employed in chapters 5 and 6. This coding scheme [22], named SESAME in the context of the MPEG4 proposal, is based on rate-constrained optimization of both the partition structure and the bit-allocation for the coding of the region contents. For a more detailed explanation for the optimization algorithm, the reader is referred to [69].

C.1 Principle and structure of the segmentation-based scheme

The segmentation algorithm does not make any assumption about the scene content, no a priori information is considered. The scene can have an arbitrary number of objects with arbitrary relations, positions and motions. This leads to a partition that should be signal dependent. Therefore, it results from an analysis of the sequence.

The representation of objects by partitions does not only involve the definition of object contours at one instant but also their time evolution. Indeed, one should be able to recognize that one region (or one object) proceeds from a given region (or object) in the previous frames. In other words, one should be able to track regions and objects in time. This approach discards all techniques that define partitions independently from one frame to another one. The coding strategy cannot rely on a fixed topology of the partition. The partition has to evolve with the modifications of the scene content: regions are to be introduced in the partition when new objects appear in the scene. Regions are to be removed when objects disappear in the scene.

The general structure of the segmentation-based coding scheme is presented in Figure C.1. The encoding process relies on three sets of functions: Partition functions, Bit allocation function and Coding functions.

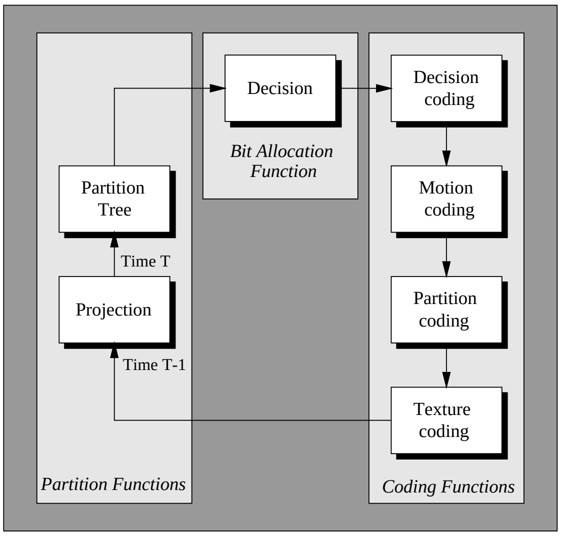


Figure C.1: General structure of the SESAME scheme

- **Partition functions** As discussed before, the sequence representation relies on signal-dependent partitions. Moreover, following the sequence evolution, regions should be tracked and the partition topology may be modified. This set of requirements is implemented by the *partition functions.* In fact, two processing steps can be distinguished (see Figure C.1): the *Projection* which tracks the time evolution of the regions, and the *Partition* tree which deals with the modifications of the partition topology (elimination and introduction of regions).

- **Bit allocation function** This function is implemented by the block called *Decision.* In order to get an efficient content- based representation, the problem of bit allocation has been carefully studied. The *Bit allocation function* optimizes the repartition between the various types of information to be coded and transmitted. In the SESAME proposal, it concerns mainly motion, partition, grey level and color information. As a result, the *Decision block* defines the coding strategy, that is, the region to be coded and the type of coding to be applied on each region. The *Decision block* has to select the best strategy in terms of regions and coding techniques among a set of possibilities. The *Decision* is made based on Rate-Distortion theory concepts and is explained in more detail in section C.2.

- **Coding functions** The last set of functions actually codes the information necessary to restore the sequence on the receiver side. They deal with the encoding of the coding strategy (Decision coding), the motion information (Motion coding), the partition (Partition coding) and the grey and color level pixel values (Texture). The partition and the texture should be motion compensated. This explains why the Motion coding block is located before the Partition and texture coding blocks.

C.2 Partition tree and rate-constrained optimization

The *partition tree* consists of a set of hierarchical partition proposals as shown in Fig. C.2. These partitions are obtained by merging and splitting regions from the projected partition (in inter-frame mode, this the partition of the current after motion compensation).

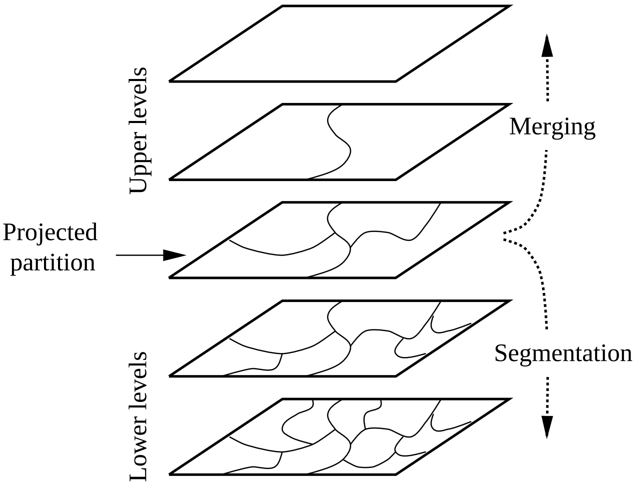


Figure C.2: Partition tree

The rate-constrained optimization algorithm decides in the *decision process* both the partition structure for each frame and the texture coding technique applied for the coding of the regions. This decision is performed on the basis of the rates and distortion values resulting from the application, both in intra-frame and inter-frame mode, of each one of the texture coding techniques to every region of the partition proposals analyzed in the partition tree. The optimization algorithm selects the optimal partition proposal and the set of coding techniques that result in the smaller distortion for a given rate. An example of optimal partition selected from the partition tree is illustrated in Fig. C.3

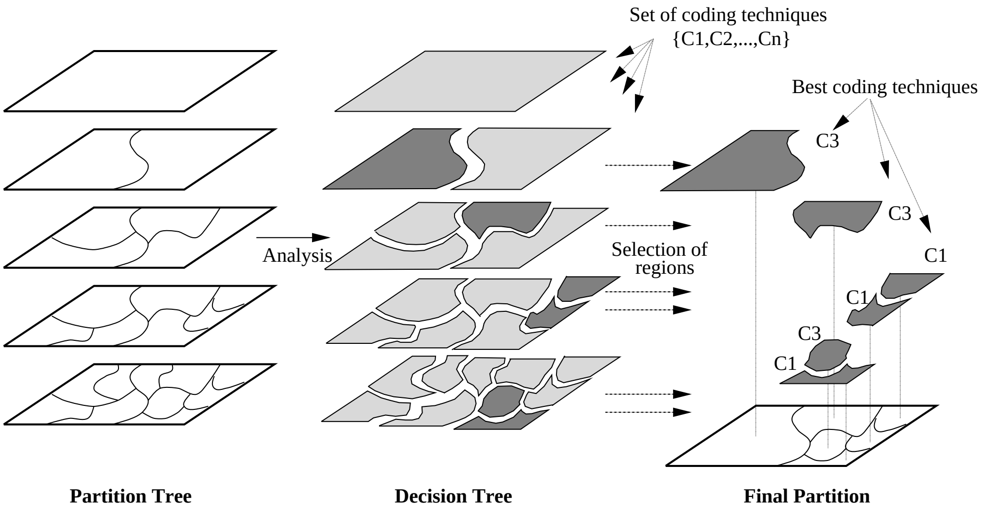


Figure C.3: Decision process

The decision process corresponds to the higher level bit-allocation algorithm mentioned in chapter 5. For a given overall target rate, the decision of the higher level algorithm results in a specific segmentation of the input image and an optimal set of rate-distortion pairs for each region. This decision is made using an minimization algorithm relying on Lagrange multipliers, which converts the constrained optimization problem of selecting the set regions and texture coding techniques that yield the smallest distortion at a given rate in an unconstrained problem of minimizing the Lagrangian cost. The first reference of bit allocation for completely arbitrary inputs and discrete quantizer sets was given by [105]. Then, the extension for more general temporally and spatially dependent coding scenarios was addressed in [83]. Finally, the application in the framework of segmentation-based coding was given in [87].

With the resulting optimal rates, the region-based subband coding bit-allocation algorithm (lower level) decides, in each region, the quantization steps for the different subbands of the frequency decomposition.

Bibliography

[1] T. Acar and Gökmen M. Image coding using weak membrane models of images. In *SPIE Visual Communications and Image Processing'94*, volume 2308, pages 1221–1230, Chicago, IL, September 1994.

[2] K. Aizawa. Model-based video coding. In L. Torres and M. Kunt, editors, *Video coding: the second generation approach*, chapter 8, pages 305–335. Kluwer Academic Publishers, Boston, 1996.

[3] K. Aizawa and T. S. Huang. Model-based image coding: advanced coding techniques for very low bit-rate applications. *Proceedings of the IEEE*, 83(2):259–271, February 1995.

[4] H. J. Barnard, J. H. Weber, and J. Biemond. Efficient signal extension for subband/wavelet decomposition of arbitrary length signals. In *SPIE Visual Communications and Image Processing'93*, volume 2094, pages 966–975, Cambridge, MA, November 1993.

[5] H. J. Barnard, J. H. Weber, and J. Biemond. A region-based discrete wavelet transform. In *EUSIPCO 94, VII European Signal Processing Conference*, pages 1234–1237, Edinburgh, U.K., September 1994.

[6] H. Benoit-Cattin, A. Baskurt, F. Peyrin, and R. Goutte. A study on FIR filters for subband coding of images. In *EUSIPCO 94, VII European Signal Processing Conference*, volume II, pages 1238–1241, Edinburgh, U.K., September 1994.

[7] L. Bouchard, J. R. Casas, B. Marcotegui, F. Meyer, C. Oddou, P. Salembier, L. Torres, and M. van Droogenbroeck. Coding of inside. Race R2053 Deliverable R2053/UPC/GPS/DS/R/007/b1, Morpheco Consortium, December 1993.

[8] L. Bouchard, E. Decenciere, and P. Salembier. From texture to feature. Race R2053 Deliverable R2053/UPC/GPS/DS/R/014/b1, Morpheco Consortium, November 1995.

[9] V. Bruce and P. R. Green. *Visual Perception.* LEA, London, 2nd edition, 1990.

[10] P. J. Burt. The pyramid as a structure for efficient computation. In A. Rosenfeld, editor, *Multiresolution Image Processing and Analysis*, chapter 2, pages 6–35. Springer-Verlag, Berlin, 1984.

[11] P. J. Burt and E. H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540, April 1983.

[12] S. Carlsson. Sketch based coding of grey level images. *EURASIP, Signal Processing*, 15(1):57–83, July 1988.

[13] J. R. Casas, P. Salembier, and L. Torres. Morphological interpolation for texture coding. In *IEEE International Conference on Image Processing*, volume I, pages 526–529, Washington DC, USA, October 1995.

[14] J. R. Casas and L. Torres. Coding of details in very-low bit-rate video systems. *IEEE Transactions on Circuits and Systems for Video Technology*, 4(3):317–327, June 1994.

[15] J. R. Casas and L. Torres. Feature-based video coding using Mathematical Morphology. In *EUSIPCO 94, VII European Signal Processing Conference*, volume I, pages 143–146, Edinburgh, U.K., September 1994.

[16] J. R. Casas and L. Torres. Morphological filter for lossless image subsampling. In *IEEE International Conference on Image Processing*, volume II, pages 903–907, Austin, TX, November 1994.

[17] J. R. Casas and L. Torres. A feature-based subband coding scheme. In *ICASSP'96*, volume IV, pages 2357–2360, Atlanta, GA, May 1996.

[18] J. R. Casas, L. Torres, and M. Jareño. Efficient coding of residual images. In *SPIE Visual Communications and Image Processing'93*, volume 2094, pages 694–705, Cambridge, MA, November 1993.

[19] B. Chitprasert and K. R. Rao. Human visual weighted progressive image transmission. *IEEE Transactions in Communications*, 38(7):1040–1044, July 1990.

[20] C.-H. Chou and Y.-C. Li. A perceptually tuned subband image coder based on the measure of just-noticeable-distortion profile. *IEEE Transactions on Circuits and Systems for Video Technology*, 5(6):467–476, December 1995.

[21] T. N. Cornsweet. *Visual Perception.* Academic Press, New York, 1970.

[22] I. Corset, L. Bouchard, S. Jeannin (LEP), P. Salembier, F. Marqués, M. Pardàs, R. Morros (UPC), F. Meyer, and B. Marcotegui (CMM). Segmentation-based coding system allowing the manipulation of objects (SESAME). Technical Report ISO/IECJTC1/SC29/WG11/MPEG95/408, LEP, UPC, CMM, November 1995.

[23] R. Crochiere and L. R. Rabiner. *Multirate Digital Filter Processing*, chapter 7, pages 289–404. Prentice-Hall, Englewood Cliffs, NJ, 1983.

[24] L. H. Croft and J. A. Robinson. Subband image coding using watershed and watercourse lines of the wavelet transform. *IEEE Transactions on Image Processing*, 3(6):759–772, November 1994.

[25] A. Cumani, P. Grattoni, and A. Guiducci. An edge-based description of color images. *Computer Vision, Graphics and Image Processing*, 53(4):313–323, July 1991.

[26] S. L. Eddins and M. J. T. Smith. A three-source multirate model for image compression. In *ICASSP'90*, volume IV, pages 2089–2092, Albuquerque, April 1990.

[27] O. Egger and W. Li. Subband coding of images using asymmetrical filter banks. *IEEE Transactions on Image Processing*, 4(4):478–485, April 1995.

[28] O. Egger, W. Li, and M. Kunt. High compression image coding using and adaptive morphological subband decomposition. *Proceedings of the IEEE*, 83(2):272–287, February 1995.

[29] P. Elias. Predictive Coding–Part I. *IRE Trans. Information Theory*, IT(2):16–33, March 1955.

[30] K. Challapali et al. The Grand Alliance System for US HDTV. *Proceedings of the IEEE*, 83(2):158–174, February 1995.

[31] D. Florêncio and R. Schafer. Critical morphological sampling and applications to image coding. In J. Serra and P. Soille, editors, *Mathematical Morphology and its Applications to Image Processing*, pages 109–116. Kluwer Academic Publishers, 1994.

[32] R. Forchheimer and O. Fahlander. Low bit-rate coding through animation. In *Picture Coding Symposium*, pages 113–114, Davis, LA, May 1983.

[33] R. Forchheimer and T. Kronander. Image coding: from waveforms to animation. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37(12):2008–2023, December 1989.

[34] H. Freeman. On the coding of arbitrary geometric configurations. *IRE Trans. Electronic Comp.*, EC-10(7):260–268, June 1961.

[35] H. Gharavi. Subband coding of video signals. In J. W. Woods, editor, *Subband Image Coding*, chapter 6, pages 229–272. Kluwer Academic Publishers, Boston, 1991.

[36] H. Gharavi and A. Tabatabai. Sub-band coding of monochrome and color images. *IEEE Transactions on Circuits and Systems*, 35(2):207–214, February 1988.

[37] M. Gilge, T. Engelhardt, and Mehlan R. Coding of arbitrarely shaped image segments based on a a generalized orthogonal transform. *EURASIP, Image Communications*, 1(2):153–180, October 1989.

[38] W. E. Glenn. Digital image compression based on visual perception and scene properties. *SMPTE Journal*, pages 392–397, May 1993.

[39] D. J. Granrath. The role of human visual models in image processing. *Proceedings of the IEEE*, 69(5):552–561, May 1969.

[40] P. Grattoni and A. Guiducci. Contour coding for image description. *Pattern Recognition Letters*, 11:95–105, February 1990.

[41] M. Grimaud. A new measure of contrast: the dynamics. In *SPIE Visual Communications and Image Processing'92*, volume 1769, pages 292–305, San Diego, CA, July 1992.

[42] Draft ITU-T Recommendation H.263. Video coding for narrow telecommunication channels at less than 64 kbit/s. Technical report, ITU, July 1995.

[43] R. M. Haralick, X. Zhuang, C. Lin, and S. J. Lee. The digital morphological sampling theorem. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37(12):2067–2090, December 1989.

[44] D. A. Huffman. A method for the construction of minimum redundancy codes. *Proceedings of the IRE*, 40(10):1098–1101, September 1952.

[45] A. K. Jain. *Fundamentals of digital image processing.* Prentice-Hall, Englewood Cliffs, NJ, 1989.

[46] N. Jayant, J. Johnston, and R. Safranek. Signal compression based on models of human perception. *Proceedings of the IEEE*, 81(10):1383–1421, October 1993.

[47] N. S. Jayant and P. Noll. *Digital coding of waveforms: principles and applications to speech and video.* Prentice-Hall, Englewood Cliffs, NJ, 1984.

[48] J. D. Johnston. A filter family designed for use in quadrature mirror filter banks. In *ICASSP'80*, Denver, USA, April 1980. IEEE.

[49] J. D. Johnston. Transform coding of audio signals using perceptual noise criteria. *IEEE Journal of Selected Areas in Communications*, pages 314–323, February 1988.

[50] G. Karlsson and M. Vetterli. Extension of finite length signals for sub-band coding. *Signal Processing*, 17(2):161–168, June 1989.

[51] S. A. Karunasekera and N. G. Kingsbury. A distortion measure for blocking artifacts in images based on human visual sensitivity. *IEEE Transactions on image processing*, 4(6):713–724, June 1995.

[52] J. J. Koenderink. The structure of images. *Biol. Cybern.*, 50:363–370, 1984.

[53] E. R. Kretzmer. Reduced-alphabet representation of television signals. *IRE Convention Record*, 4:140–147, 1956.

[54] M. Kunt, A. Ikonomopoulos, and M. Kocher. Second generation image coding techniques. *Proceedings of the IEEE*, 73(4):549–575, April 1985.

[55] O.-J. Kwon and R. Chellappa. Region-based subband image coding scheme. In *IEEE International Conference on Image Processing*, volume II, pages 859–863, Austin, TX, November 1994.

[56] D. Le Gall. The MPEG video compression algorithm. *Signal Processing: Image Communications*, 4:129–140, April 1992.

[57] D. Le Gall and A. Tabatabai. Subband coding of digital images using symmetric short kernel filters and arithmetic coding techniques. In *ICASSP'88*, volume V, pages 761–764, New York, April 1988.

[58] H. Li. *Low bit-rate image sequence coding.* PhD thesis, Dept. of Electrical Engineering, Linköping University, Linköping, Sweden, 1993.

[59] H. Li, A. Lundmark, and R. Forchheimer. Image sequence coding at very low bit-rates: a review. *IEEE Transactions on Image Processing*, 3(5):589–609, September 1994.

[60] J. S. Lim. *Two-dimensional Signal and Image Processing.* Prentice-Hall, Englewood Cliffs, N. J., 1990.

[61] M. L. Liou. Visual telephony as an ISDN application. *IEEE Communications Magazine*, 28(2):30–38, February 1990.

[62] J. L. Mannos and D. L. Sakrison. The effect of a visual fidelity criterion on the encoding of images. *IEEE Transactions on Information Theory*, 20(4):525–536, July 1974.

[63] J. B. Mariño, F. Vallverdú, J. A. Rodríguez Fonollosa, and A. Moreno. *Tratamiento Digital de la Señal. Una Introducción Experimental.* Politext. Edicions UPC, Barcelona, 1995.

[64] F. Marqués. *Multiresolution image segmentation based on compound random fields. Application to image coding.* PhD thesis, Dept. of Signal Theory and Communications, UPC, Barcelona, December 1992.

[65] F. Marqués, J. Sauleda, and A. Gasull. Shape and location coding for contour images. In *Picture Coding Symposium*, pages 18.6.1–18.6.2, Lausanne, Switzerland, March 1993.

[66] D. Marr. *Vision.* Freeman, New York, 1982.

[67] F. Meyer. Contrast feature extraction. In J. L. Chermant, editor, *Quantitative Analysis of Micro-structures in Materials Sciences, Biology and Medicine.* Riederer Verlag, Stuttgart, 1978.

[68] F. Meyer. Morphological image segmentation for coding. In J. Serra and P. Salembier, editors, *Proceedings of the First Workshop on Mathematical Morphology and its Applications to Signal Processing*, pages 46–51, Barcelona, Spain, May 1993. UPC.

[69] R. Morros, F. Marqués, M. Pardàs, and P. Salembier. Video sequence segmentation based on rate-distortion theory. In *SPIE Visual Communication and Image Processing'96*, volume 2727, Orlando, FL, March 1996.

[70] MPEG. MPEG-4 Proposal Package Description (PPD). Technical Report ISO/IEC JTC1/SC29/WG11, MPEG, July 1995.

[71] A. N. Netravali and B. G. Haskell. *Digital pictures: representation and compression.* Plenum, New York, 1988.

[72] A. N. Netravali and J. O. Limb. Picture coding: a review. *Proceedings of the IEEE*, 68(3):366–406, March 1980.

[73] A. N. Netravali and B. Prasada. Adaptive quantization of picture signals using spatial masking. *Proceedings of the IEEE*, 65(4):536–548, April 1977.

[74] A. V. Oppenheim and R. W. Schafer. *Discrete-time Signal Processing.* Prentice-Hall, Englewood Cliffs, NJ, 1989.

[75] M. Pardàs, P. Salembier, F. Marqués, and R. Morros. Partition tree for segmentation-based video coding. In *ICASSP'96*, Atlanta, GA, May 1996.

[76] D. E. Pearson. Visual communication systems for the deafs. *IEEE Transactions on Communications*, 29(12):1986–1992, December 1981.

[77] D. E. Pearson and J. A. Robinson. Visual communications at very low data rates. *Proceedings of the IEEE*, 73(4):795–812, April 1985.

[78] S.-C. Pei and F.-C. Chen. Subband decomposition of monochrome and color images by Mathematical Morphology. *Optical Engineering*, 30(7):921–933, July 1991.

[79] S.-C. Pei and F.-C. Chen. 3-D spatio-temporal subband decomposition for hierarchical compatible video coding by Mathematical Morphology. *Signal Processing: Image Communications*, 6(1):83–99, March 1994.

[80] M. G. Perkins and T. Lookabaugh. A psychophysically justified bit allocation algorithm for subband image coding systems. In *ICASSP'89*, volume 3, pages 1815–1818, Glasgow, Scotland, May 1989.

[81] C. Podilchuk and N. Farvardin. Perceptually based low bit-rate video coding. In *ICASSP'91*, volume 4, pages 2837–2840, Toronto, Canada, May 1991.

[82] W. K. Pratt. *Digital Image Processing*. Wiley, New York, 2nd edition, 1991.

[83] K. Ramchandran, A. Ortega, and M. Vetterli. Bit allocation for dependent quantization with aplications to multiresolution and MPEG video coders. *IEEE Transactions on Image Processing*, 3(5):533–545, September 1994.

[84] K. Ramchandran and M. Vetterli. Best wavelet packet bases in a rate-distorsion sense. *IEEE Transactions on Image Processing*, 2(2):160–175, April 1993.

[85] X. Ran and N. Farvardin. A perceptually motivated three-component image model. Part I: Description of the model. *IEEE Transactions on Image Processing*, 4(4):401–415, April 1995.

[86] X. Ran and N. Farvardin. A perceptually motivated three-component image model. Part II: Application to image compression. *IEEE Transactions on Image Processing*, 4(4):430–447, April 1995.

[87] E. Reusens. Joint optimization of representation model and frame segmentation for generic video compression. *EURASIP Signal Processing*, 46(11):105–117, September 1995.

[88] O. Rioul. On the choice of wavelet filters for still image compression. In *ICASSP'93*, volume V, pages 550–553, Minneapolis, MN, April 1993.

[89] O. Rioul and M. Vetterli. Wavelets and signal processing. *IEEE Signal Processing Magazine*, 8(4):14–38, October 1991.

[90] J. A. Robinson. Image coding with ridge and valley primitives. *IEEE Transactions on Communications*, 43(6):2095–2102, June 1995.

[91] J. A. Robinson and M. S. Ren. Data-dependent sampling of two-dimensional signals. *Multidimensional Systems and Signal Processing*, 6:89–111, February 1995.

[92] R. J. Safranek and J. D. Johnston. A perceptually tuned subband image coder with image dependent quantization and post-quantization data compression. In *ICASSP'89*, volume 3, pages 1945–1948, Glasgow, Scotland, May 1989.

[93] P. Salembier and M. Kunt. Size-sensitive multiresolution decomposition with rank order based filters. *EURASIP, Signal Processing*, 27(2):205–241, May 1992.

[94] P. Salembier and M. Pardàs. Hierarchical morphological segmentation for image sequence coding. *IEEE Transactions on Image Processing*, 3(5):639–651, September 1994.

[95] P. Salembier and R. Rué. Texture coding using morphological interpolation. In IEEE, editor, *1995 IEEE Workshop on Nonlinear Signal and Image Processing*, pages 258–261, Halkidiki, Greece, June 20-22 1995.

[96] P. Salembier and J. Serra. Morphological multiscale image segmentation. In *SPIE Visual Communication and Image Processing'92*, volume 1818, pages 620–631, Boston, MA, November 1992.

[97] P. Salembier, L. Torres, F. Meyer, and C. Gu. Region-based video coding using mathematical morphology. *Proceedings of IEEE (Invited Paper)*, 83(6):843–857, June 1995.

[98] C. P. Sandbank. *Digital Television.* Wiley, Chichester, 1990.

[99] H. Sanson. Towards a robust parametric identification of motion on regions of arbitrary shape by non-linear optimization. In *IEEE International Conference on Image Processing*, volume I, pages 203–206, Washington DC, USA, October 1995.

[100] G. Sapiro. Geometric partial differential equations in image analysis: past, present, and future. In *IEEE International Conference on Image Processing*, volume III, pages 1–4, Washington DC, USA, October 1995.

[101] W. F. Schreiber. The design of improved television systems. In *Fundamentals of Electronic Imaging Systems: Some Aspects of Image Processing*, chapter 8. Springer-Verlag, Berlin, 2nd edition, 1991.

[102] W. F. Schreiber, C. F. Knapp, and N. D. Kay. Synthetic highs, an experimental TV bandwidth reduction system. *SMPTE Journal*, 68:525–537, August 1959.

[103] J. Serra. *Image Analysis and Mathematical Morphology.* Academic Press, London, 1982.

[104] J. Serra. *Image Analysis and Mathematical Morphology, Vol II: Theoretical advances.* Academic Press, London, 1988.

[105] Y. Shoham and A. Gersho. Efficient bit allocation for an arbitrary set of quantizers. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(9):1445–1453, September 1988.

[106] R. Sibson. A brief description of natural neighbor interpolation. In Vic Barnett, editor, *Interpreting Multivariate Data*, chapter 2, pages 21–36. Wiley, Chichester, 1981.

[107] T. Sikora and B. Makai. Shape-Adaptive DCT for generic coding of video. *IEEE Transactions on Circuits and Systems for Video Technology*, 5(1):59–62, February 1995.

[108] M. J. T. Smith. IIR analysis/synthesis systems. In J. W. Woods, editor, *Subband Image Coding*, chapter 3, pages 101–141. Kluwer Academic Publishers, Boston, 1991.

[109] M. J. T. Smith and T. P. Barnwell, III. Exact reconstruction techniques for tree-structured subband coders. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 34(3):434–441, June 1986.

[110] P. Soille. Spatial distributions from contour lines: an efficient methodology based on distance transformations. *Journal of Visual Communication and Image Representation*, 2(2):138–150, June 1991.

[111] F.-K. Sun and P. Maragos. Experiments on image compression using morphological pyramids. In *SPIE Visual Communications and Image Processing'89*, volume 1199, pages 1303–1312, 1989.

[112] ISO-IEC CD 13818 Information Technology. Generic coding of moving pictures and associated audio (MPEG-2). Technical report, Motion Picture Expert Group, November 1993.

[113] A. Toet. A morphological pyramidal image decomposition. *Pattern Recognition Letters*, 9:255–261, May 1989.

[114] K. M. Uz, M. Vetterli, and D. J. LeGall. Interpolative multiresolution coding of advanced television with compatible subchannels. *IEEE Transactions on Circuits and Systems for Video Technology*, 1(1):86–99, March 1991.

[115] P. P. Vaidyanathan. Quadrature mirror filter banks, M-band extensions and perfect-reconstruction techniques. *IEEE ASSP Magazine*, 4, July 1987.

[116] L. J. van Vliet, I. T. Young, and G. L. Beckers. A nonlinear laplace operator as edge detector in noisy images. *Computer Vision, Graphics and Image Processing*, 45:167–195, 1989.

[117] D. Vernon. *Machine Vision.* Prentice-Hall, London, 1991.

[118] M. Vetterli. Multi-dimensional subband coding: some theory and algorithms. *Signal Processing*, 6(2):97–112, April 1984.

[119] M. Vetterli. Multirate filter banks for subband coding. In J. W. Woods, editor, *Subband Image Coding*, chapter 2, pages 43–100. Kluwer Academic Publishers, Boston, 1991.

[120] L. Vincent. Morphological gray scale reconstruction in image analysis: applications and efficients algorithms. *IEEE, Transactions on Image Processing*, 2(2):176–201, April 1993.

[121] L. Vincent and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE, Transactions on Pattern Analyis and Machine Intelligence*, 39(12):1845–1855, December 1991.

[122] G. K. Wallace. The JPEG still picture compression standard. *Communications of the ACM*, 34(4):30–44, April 1991.

[123] Y. Wang and O. Lee. Active mesh - a feature seeking and tracking image sequence representation scheme. *IEEE Transactions on Image Processing*, 3(5):610–624, September 1994.

[124] P. H. Westerink, J Biemond, and D. E. Boekee. An optimal bit allocation algorithm for subband coding. In *ICASSP'88*, pages 757–760, New York, April 1988.

[125] P. H. Westerink, J Biemond, and D. E. Boekee. Subband coding of color images. In J. W. Woods, editor, *Subband Image Coding*, chapter 5, pages 193–227. Kluwer Academic Publishers, Boston, 1991.

[126] I. H. Witten, R. M. Neal, and J. G. Cleary. Arithmetic coding for data compression. *Communications of the ACM*, 30(6):520–540, June 1987.

[127] J. W. Woods. *Subband Image Coding.* Kluwer Academic Publishers, Boston, 1991.

[128] J. W. Woods and S. D. O'Neil. Subband coding of images. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(5):1278–1288, October 1986.

[129] J. K. Yan and D. J. Sakrison. Encoding of images based on a two-component source model. *IEEE Transactions on Communications*, 25(11):1315–1322, November 1977.

[130] Y. Yasuda. Overview of digital facsimile coding techniques in Japan. *Proceedings of the IEEE*, 68(7):830–845, July 1980.

[131] S. Zhang, M. Liang, J. A. Robinson, and G. L. Greig. Motion coding of spatial primitives. *Signal Processing: Image Communication*, 7(4-6):457–469, November 1995.

[132] Z. Zhou and A. N. Venetsanopoulos. Morphological methods in image coding. In *ICASSP'92*, volume III, pages 481–484, San Francisco, CA, March 1992.

Agraïments

Vull agrair molt sinceramment la col.laboració de totes les persones
que han fet possible que aquesta tesi arribi a bon port.

A en Lluís Torres primer, per acceptar-ne la direcció
i per empentar-me quan jo frenava.

Al tots companys del grup d'imatge. Especialment a en Philippe,
que m'ha rebut en totes les meves anades i vingudes
i m'ha donat tot un suport tècnic inestimable.

Als amics de sempre, que han soportat estoicament la meva vida monàstica
i, després de tot aquest temps, encara em parlen.

Vull fer una menció especial d'en Jean Serra. La seva profunda perspectiva
i àmplia visió de les coses és aclaparadora
Ell ha estat la causa (involuntària?) de la meva situació (personal).
Li agraeixo profundament les interessants discussions que vàrem mantenir
durant la seva estada a Barcelona i, sobretot, les oportunitats que em va donar.

A en Fernand Meyer, li estic en deute per molts motius.
Una gran part del que hi ha escrit aquí al davant
ho va suggerir ell o el seu treball.

A la Bea,
sempre disposada a enviar-me els papers que li demanava.

I com no.
A qui ha esperat aquest moment amb més ànsia encara que no pas jo.
A qui m'ha compartit durant gairebé quatre anys amb una noia anomenada
"Tesi",
que m'acaparava durant gairebé totes les petites estones lliures.
A qui, conscientment, m'hi ha deixat fer les meves escapades de cap de setmana
altruïstament i sense queixar-se massa.
A qui, en definitiva, dedicaré el cent u per cent de la meva vida a partir d'ara.
A l'Àngels.

Per acabar, vull dedicar aquest treball, a la meva família,
sobre tot, als meus pares, que han fet que 'tot' hagi pogut ser.

Març de 1996.