

# CHAPTER 5

## Three-Dimensional Subband Coding

---

### 5.1 Three-dimensional subband analysis and synthesis

#### 5.1.1 Implementation with perfect reconstruction filter banks

#### 5.1.2 Discussion of the implementation

### 5.2 Encoding of the subbands

#### 5.2.1 DPCM encoding of the base band

#### 5.2.2 PCM encoded bands

#### 5.2.3 Run-length encoding

### 5.3 Complexity and parallelism

### 5.4 Packetization and error concealment

### 5.5 Extensions to the coding scheme

### 5.6 Summary

---

A digital video signal is three-dimensional, with two dimensions in space and the third over time. We wish the coding to remove redundancy in each dimension, while at the same time, the coding implementation has to incorporate the aspects outlined in Chapter 2. Therefore, a suitable coding scheme has to be evaluated both for its ability to compress the video signal as well as its ability to assure transmission over the network in a loaded condition, where packet loss and delays are non-negligible. We have found that subband coding is well suited as hierarchical coding for packet video. The three-dimensional subband coding method that we propose gives good integration with the transmission aspects, high compression and quality, and it may be implemented in parallel at low complexity. Moreover, the theoretical results of the previous chapter provide a foundation for further evolution of the coding method. However, the implementation described in this chapter relies on separability of both the subsampling lattice and the filter banks.

These simplifications were accepted primarily for the readily available perfect reconstruction filters [LEG88b], but separability also facilitates implementation and it yields a lower complexity than a non-separable system. A related study, pertaining to extension of finite length signals for subband coding, is presented in Appendix B. The findings of that study are incorporated into the implementation described in this chapter.

### 5.1 Three-dimensional subband analysis and synthesis

In the previous chapter, we derived general conditions for subband coding which are valid for any two-dimensional geometry. Here we will focus on a restricted case where the subsampling lattice is rectangular (*i.e.*,  $d_{01} = 0$  in Eq. (4.1)), as the input lattice, and the filter banks are separable. By restricting the geometry of the subsampling and by considering only separable filter banks, the system may, however, be more easily extended to encompass the three-dimensions of video signals. It also simplifies iterated subband analysis of one or more subbands into yet smaller frequency regions.

#### 5.1.1 Implementation with perfect reconstruction filter banks

The dimensionality of previously presented subband coding implementations (see for example [WOO86, SMI87b, LEG88b, WES88a, GHA88]) is extended to video by including subband analysis in the temporal direction. Since the system is separable, the analysis filter bank may be written as

$$\bar{H}(z_1, z_2, z_3) = \bar{H}_t(z_1) \otimes \bar{H}_h(z_2) \otimes \bar{H}_v(z_3). \quad (5.1)$$

The  $\bar{H}$ 's on the right-hand side are column-vectors where each element is impulse-response of a filter, and their subscripts  $t$ ,  $h$  and  $v$  stand for the temporal direction,

the horizontal and vertical spatial directions, respectively. The operator, ' $\otimes$ ', is the Kronecker matrix product. Consequently, the directional filter banks,  $\bar{H}_t(z_1)$ ,  $\bar{H}_h(z_2)$  and  $\bar{H}_v(z_3)$ , are one-dimensional, although the system is three-dimensional. Therefore the implementation may rely on existing well-performing one-dimensional filter banks.

In the following, we concentrate on an analysis which yields two subbands in each direction, with a total of eight bands. We write this system as

$$\bar{H}(z_1, z_2, z_3) = \begin{pmatrix} H_{t0}(z_1) \\ H_{t1}(z_1) \end{pmatrix} \otimes \begin{pmatrix} H_{h0}(z_2) \\ H_{h1}(z_2) \end{pmatrix} \otimes \begin{pmatrix} H_{v0}(z_3) \\ H_{v1}(z_3) \end{pmatrix}, \quad (5.2)$$

where filter index '0' indicate a low-pass filter and '1' a high-pass filter. The corresponding synthesis filter bank,  $\bar{G}(z_1, z_2, z_3)$ , is defined analogously. The subsampling and the following upsampling, both by a factor of 2 in each dimension, yield the analysis filtered signal modulated by

$$f(n_v, n_h, n_t) = \frac{1}{8} [1 + (-1)^{n_v}] [1 + (-1)^{n_h}] [1 + (-1)^{n_t}], \quad (5.3)$$

where the  $n$ 's are the sample indices. In other words, one sample is retained out of every cell of  $2 \times 2 \times 2$  samples, and the other samples have been discarded by the subsampling and reinserted as zero-valued samples by the upsampling. For the two-band case we thus have the input/output relation:

$$\begin{aligned} \hat{X}(z_1, z_2, z_3) = & \frac{1}{8} \bar{G}(z_1, z_2, z_3)^T [\bar{H}(z_1, z_2, z_3) X(z_1, z_2, z_3) \\ & + \bar{H}(-z_1, z_2, z_3) X(-z_1, z_2, z_3) + \bar{H}(z_1, -z_2, z_3) X(z_1, -z_2, z_3) \\ & + \bar{H}(-z_1, -z_2, z_3) X(-z_1, -z_2, z_3) + \bar{H}(z_1, z_2, -z_3) X(z_1, z_2, -z_3) \\ & + \bar{H}(-z_1, z_2, -z_3) X(-z_1, z_2, -z_3) + \bar{H}(z_1, -z_2, -z_3) X(z_1, -z_2, -z_3) \\ & + \bar{H}(-z_1, -z_2, -z_3) X(-z_1, -z_2, -z_3)] \end{aligned} \quad (5.4)$$

The aliasing components are nonzero when the frequency responses of the analysis filters overlap. Aliasing distortion is highly visible in images but can be canceled by the following constraints on the synthesis filters:

$$G_{d0}(z) = H_{d1}(-z), \quad \text{and} \quad G_{d1}(z) = -H_{d0}(-z), \quad (5.5)$$

where  $d$  denote direction (*i.e.*,  $v$ ,  $h$ , and  $t$ ). The aliasing is now perfectly canceled regardless of the amount of overlap of the passbands of the filters. Hence, we are left with

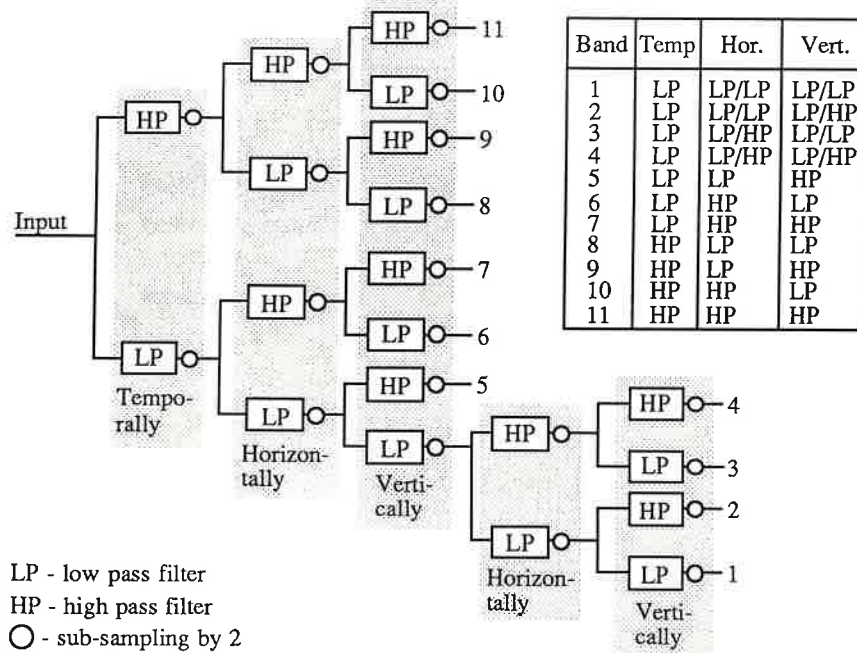
$$\begin{aligned} \hat{X}(z_1, z_2, z_3) &= \frac{1}{8} \bar{G}(z_1, z_2, z_3)^T \bar{H}(z_1, z_2, z_3) X(z_1, z_2, z_3) \\ &= \frac{1}{8} [H_{t0}(z_1) H_{t1}(-z_1) + H_{t1}(z_1) H_{t0}(-z_1)] \\ &\quad [H_{h0}(z_2) H_{h1}(-z_2) + H_{h1}(z_2) H_{h0}(-z_2)] \\ &\quad [H_{v0}(z_3) H_{v1}(-z_3) + H_{v1}(z_3) H_{v0}(-z_3)] X(z_1, z_2, z_3). \end{aligned} \quad (5.6)$$

It is desirable to chose the filters so that  $\hat{X}(z)$  is a perfect, but possibly delayed, replica of the input signal  $X(z)$  in the absence of coding and transmission loss. This can be written as

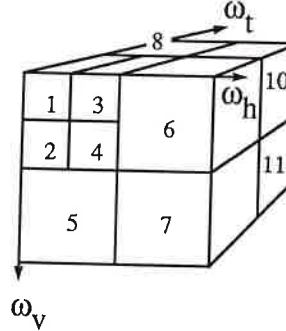
$$\hat{X}(z_1, z_2, z_3) = z_1^{-k} z_2^{-l} z_3^{-m} X(z_1, z_2, z_3), \quad \text{where } k, l, m \in \mathcal{N}. \quad (5.7)$$

To increase compression, we shall not satisfy ourselves with the discussed eight-band decomposition. The first subband, which is yielded by  $H_{t0}(z_1) H_{h0}(z_2) H_{v0}(z_3)$ , will be split spatially into four new bands. Next we present this implemented and simulated eleven-band subband coder.

The implementation of the three-dimensional subband analysis system is shown in Fig. 5.1. The system consists of temporal, horizontal, and vertical filtering, with further spatial analysis of the subband that has been obtained through low pass filtering in all of the three first stages. The resulting frequency regions are illustrated



*Figure 5.1 A three-dimensional sub-band analysis system. The subsampling is performed in the same direction as the filtering operation that it follows.*



*Figure 5.2 The 11 frequency regions of the subband analysis. The total region is initially split along the mid-points of each frequency axis. The one which contains both low temporal and low spatial frequencies, is split into four spatial-frequency regions.*

by the three-dimensional frequency diagram in Fig. 5.2. Note that owing to the subsampling the bit rate in each of the output branches from an analysis stage is half of its input rate. Consequently, the first stage with temporal filters has the highest computational burden since there is no parallelism, while each of the spatial filters operates in parallel at a lower rate. Hence, we chose the temporal filters to

be the shortest possible, which also minimizes the number of frames needed to be stored as well as the delay. In the  $z$ -transform domain the analysis filters are given by

$$H_{t0}(z) = \frac{1}{2}(1 + z^{-1}), \quad H_{t1}(z) = \frac{1}{2}(1 - z^{-1}), \quad (5.8)$$

and the synthesis filters can be easily found by Eq. (5.5). Note that these filter just correspond to a DFT of length 2.

The spatial filters operate in parallel at a lower rate, and the filter lengths do not affect the storage requirements. Consequently, we can allow higher order filters. Le Gall [LEG88b] has derived pairs of perfect reconstruction filters that are well suited for image processing. These filters have linear phase, low computational complexity, and relatively good characteristics for frequency selection and interpolation. From among them we have chosen the following filters for the spatial subband analysis and synthesis ( $h$  and  $v$  stand for horizontal and vertical, respectively):

$$H_{(h,v)0}(z) = \frac{1}{4}(-1 + 2z^{-1} + 6z^{-2} + 2z^{-3} - z^{-4}), \quad (5.9)$$

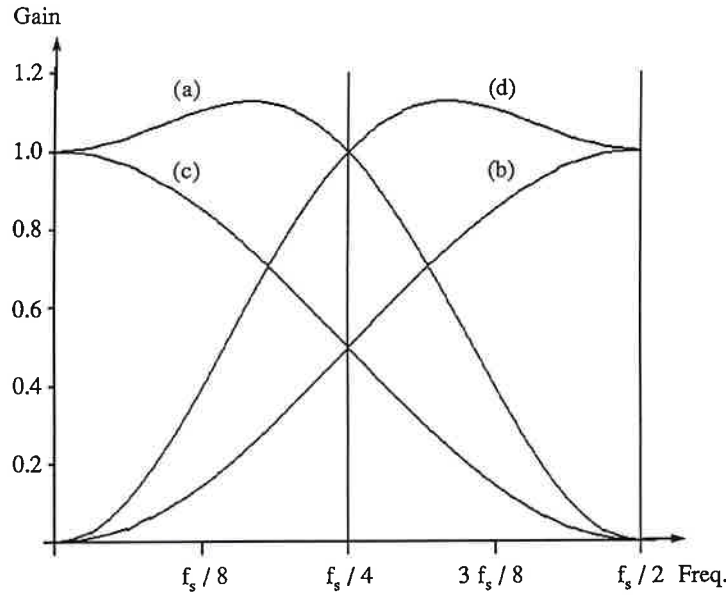
$$H_{(h,v)1}(z) = \frac{1}{4}(1 - 2z^{-1} + z^{-2}), \quad (5.10)$$

$$G_{(h,v)0}(z) = \frac{1}{4}(1 + 2z^{-1} + z^{-2}), \quad (5.11)$$

$$G_{(h,v)1}(z) = \frac{1}{4}(1 + 2z^{-1} - 6z^{-2} + 2z^{-3} + z^{-4}). \quad (5.12)$$

As can be verified, for the given filters in Eqs. (5.8) to (5.12), the input/output relationship, in Eq. (5.6), reduces to that of Eq. (5.7), with the delays  $k = 1$ , and  $l = m = 3$ . Note, however, that the spatial filters are not paraunitary. The frequency responses of the four spatial filters are shown in Fig. 5.3. The low pass filter of shorter length,  $G_{(h,v)0}$ , given in Eq. (5.11), has been chosen as the synthesis low pass filter because of its better interpolation characteristics. All filters are chosen

with regard to their low computational complexity, which is discussed in Section 5.3.



**Figure 5.3** Frequency responses of the spatial filters given by Eqs. (5.9) to (5.12): (a)  $\frac{1}{2}|H_{(h,v)0}(e^{-j2\pi f})|$ , (b)  $|H_{(h,v)1}(e^{-j2\pi f})|$ , (c)  $|G_{(h,v)0}(e^{-j2\pi f})|$ , and (d)  $\frac{1}{2}|G_{(h,v)1}(e^{-j2\pi f})|$  ( $f_s$  denotes the sampling frequency).

A complication in dealing with sub-band coding of finite length signals pertains to the signal extension, which is necessary for the filtering operations. Since an image is of finite size, the convolutions for the filterings need support outside the image (a border with the width of approximately half the filter length). The two main issues are how to extend the input to avoid distorting the signal through the analysis and synthesis stages and, under the use of lossy compression, not affect the compressability of the signal. The latter comes about if the boundary values create a discontinuity so the higher frequency components yield artificially high energy at the boundaries. This may yield considerable distortion when highly compressed. In a study reported in the Appendix, we have found that signal extension through replication of the edge values of an image is sufficient to cancel all visible distortion

at the image boundaries, while not adding to the complexity of the system. Thus, all frames of the video signal are extended this way horizontally and vertically both at the analysis and the synthesis. Finally, note that it is not possible to shift the filters to achieve a zero-delay transfer function. However, for the spatial dimensions of an image sequence, the delay for these filters can be minimized to one sample interval per analysis and synthesis stage.

### 5.1.2 Discussion of the implementation

Through the subband analysis we have achieved an hierarchical separation of the input data into 11 parallel bands. Each band has a known importance in terms of image features and quality, to which the encoding can be tailored. Thus, masking properties of the human visual system may be incorporated to yield good image quality while the distortion may be high in a mean square error sense. The base band (band 1), which has been low pass filtered through all stages, is the most vital, and the importance of the other bands decreases with increasing band index, loosely speaking (see Fig. 5.1 for the band indices). This can be seen in Fig. 5.4, where the subband analysis of Fig. 5.1 is performed on an image sequence. In the center of the top row is the input, with the temporal low pass component on the left and the temporal high pass component on the right. Underneath, from left to right, are the subsampled horizontal low and high pass components of the respective components above. In the third row from the top, the result of vertical analysis and subsampling is shown, as applied to the components above. The components are alternately low and high pass, starting with a low pass filtered component at the left. The four pictures in the last row represent bands 1 to 4, which have been obtained by horizontal and vertical analysis of the leftmost picture in the third row.

Temporal decomposition allows us to exploit the vast amount of redundancy in



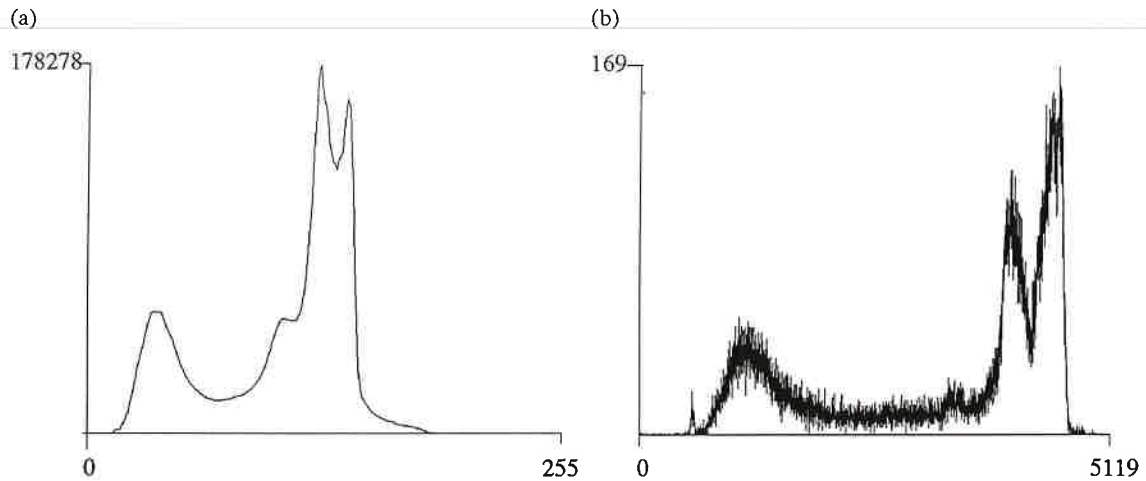


*Figure 5.4 Three-dimensional subband analysis of a video sequence.*

the time dimension without resorting to interframe prediction, which is usually not as robust to transmission error. This decomposition yields one component with essentially constant bit rate when encoded, while the other component exhibits a bursty behavior after encoding. The second stage of spatial analysis filtering, which yields bands 1 to 4, is necessary to achieve high compression. The component resulting from the low pass analysis filterings of the first three stages requires sophisticated encoding for its redundancy reduction, analogously to the input sequence, which it resembles. Thus, in accordance with the supporting arguments given for subband coding of the original, the solution for this band's coding is further spatial analysis into four more bands.

Note that seven of the bands each have only  $1/8$  of the input data rate, and the remaining four have only  $1/32$  of the input rate. Moreover, for the purpose of parallel implementation, we will treat the subbands as being independent. Thus,

all subsequent processing, such as error control encoding and packetization, may be done in parallel at more tractable data rates.



*Figure 5.5 Histograms of the intensity distribution (a) of an input sequence and (b) of subband 1, computed over 30 frames of the sequence “Miss B” (see Chapter 6). Note that band 1 has an expanded range of intensity values compared to the 256 permissible levels of the input. This is due to the gain factor of the spatial analysis low pass filter (see Eq. (5.9)).*

## 5.2 Encoding of the subbands

The subband analysis has not resulted in any compression: the sum of samples in the subbands equals that of the input. However, it has yielded an attractive separation of the data. The filtering operations yield one band (band 1) with an intensity distribution similar to that of the input (compare Figs. 5.5 (a) and (b)). The other 10 bands (bands 2 to 11) have distributions highly concentrated at or around zero and variance highly reduced compared to that of the input distribution, as shown for some of the bands in Figs. 5.6 (a) to (d). They may therefore be quantized to a reduced number of intensity levels without introducing high amounts of visible distortion. Band 1 is still highly correlated in all dimensions since it is similar to the input. This correlation is partially removed by DPCM encoding. Note that any

method of image coding could be used for this purpose, for example discrete cosine transform [LEG88a]. (Fig. 5.1 gives the band indices.)

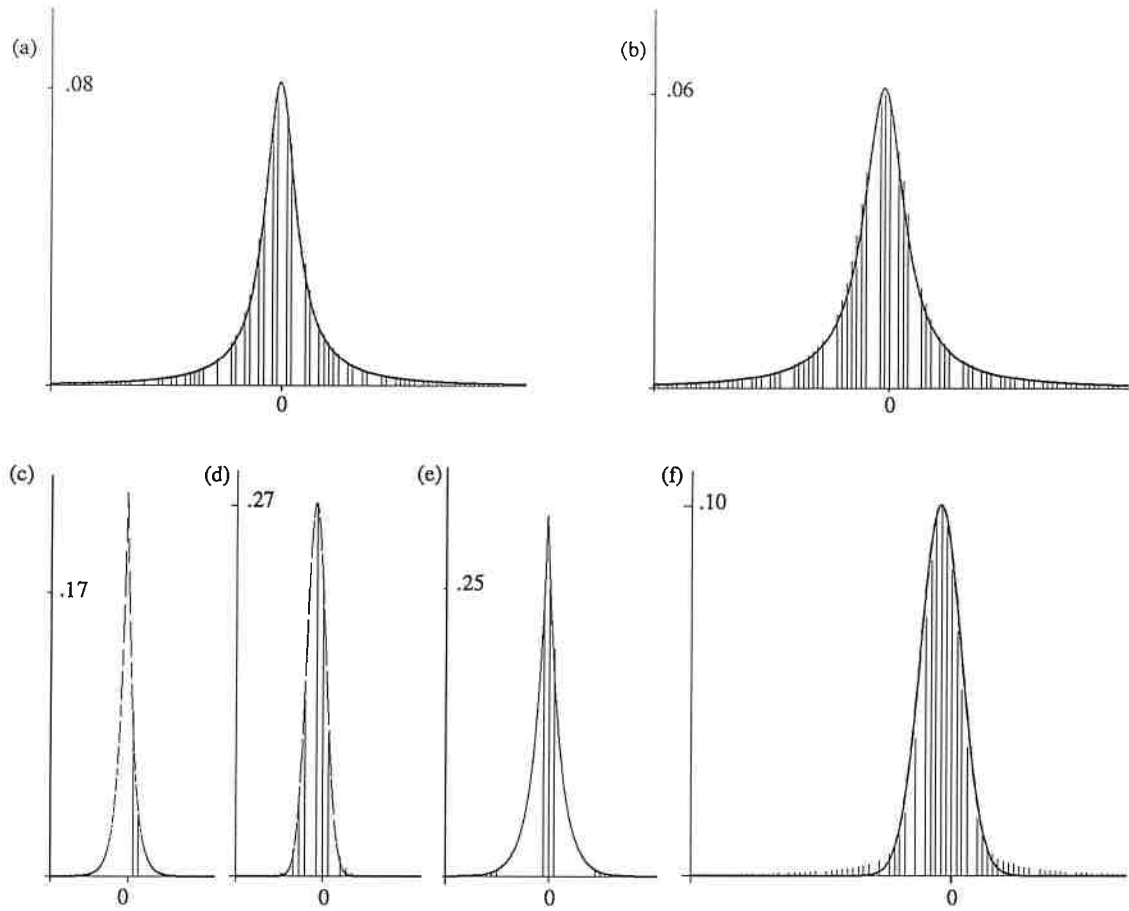
Quantization of the pixel values and the prediction error, respectively, results in large connected areas of zero-valued pixels in each band. To boost the compression, only non-zero values are transmitted and their locations are given by a run-length code.

### 5.2.1 DPCM encoding of the base band

Subband 1, which is obtained through spatial low pass filtering of the temporal low pass component, is the subband that carries most of the information about the original image. Because of its importance it will be transmitted with a higher priority than the other subbands. Although this band contains only  $1/32$  the amount of data in the input, it still has too high a data rate for requesting high priority transmission. We encode this band with a simple one-dimensional DPCM scheme. In our case we found vertical prediction slightly superior to horizontal prediction in terms of compression. Although the signal exhibits strong correlation over all three dimensions, the issue of robustness and error recovery is too important to allow the introduction of dependencies in more than one dimension. Again for the sake of robustness, spatial prediction is preferred over temporal prediction; thereby the effects of transmission loss are confined to the frame in which the loss occurs. The interframe redundancy may be well exploited for error recovery (see Section 5.4), so the prediction loop does not have to be made leaky, which would reduce the compression.

The quantization of the prediction error signal has to be done with caution since, through the synthesis, each pixel will be upsampled and interpolated to a  $4 \times 4$  block

of pixels in each of two frames. This may accentuate the visibility of artifacts, such as contouring. This subband offers the smooth form from which the other bands carve out features and detail. Consequently, contouring is the foremost artifact to be guarded against, followed by a lowered dynamic range; both are due to improper quantization. Henceforth, the quantizer has to have enough steps to allow fine quantization while still covering the dynamic range of the prediction error.



*Figure 5.6 Histograms of the intensity distribution of six subbands: (a) band 2, (b) band 3, (c) band 4, (d) band 5, (e) band 6, and (f) band 8. The best fit of the probability density functions given by Eq. (5.13) to (5.15) is overlaid on each histogram (see Table 5.1). The values on the vertical axes are the peak values of the histograms.*

In our case, the prediction errors are quantized by a symmetric, uniform quantizer.

The quantizer has been designed with virtually unlimited range, whereby only the width of the zero level and the step size have to be adjusted to fit the quantizer to the band's properties. The representative levels are taken as the mid-point between two quantization levels. These output levels are assigned variable length codewords which are fitted to an exponential distribution. Thus, the outer limit of the quantizer is set to the point where the length of the codeword would be impractically long. It has been shown beneficial to allow a quantizer with a wide range, which relieves the problem of choosing between step size and number of quantization levels. The perceptual importance of the outer levels is higher than is expected by their rare occurrence. The locations of the stretches of zero values and nonzero values, respectively, are run-length encoded, as described in Section 5.2.3.

### 5.2.2 PCM encoded bands

Through the subband division we have obtained 11 bands that we may treat as independent, of which 10 have a much lower variance in the intensity distribution than does the original signal. Hence, for these bands little can be gained through predictive coding by exploiting the already low amounts of correlation, and therefore these 10 bands are PCM encoded. The distributions of some bands are shown in Fig. 5.6. Note that the components that have been spatially low pass filtered in any direction have a larger spread of the distribution. This is not necessarily due to a higher information content, but to the higher gain factor of the spatial low pass filter, compared to the spatial high pass filter (*confer* Eqs. (5.9) and (5.10)). The quantizers for these components can be designed to offset this without requiring more levels.

We have attempted to fit probability density functions to the histograms of intensity values for each subband. The candidate functions are [PAP84]:

Subband	$\mu$	$\sigma$	Gauss	Laplace	Cauchy
2	-0.36	13.5	43.7	18.4	11.3
3	-1.21	16.3	37.0	16.6	10.9
4	-0.38	4.1	33.7	28.1	47.0
5	-0.58	4.8	71.8	67.8	81.5
6	-1.12	1.9	14.6	70.3	87.1
7	-0.37	0.7	6.3	93.5	160.7
8	-2.15	10.9	29.2	32.8	35.8
9	-0.46	3.0	36.0	32.3	63.9
10	-1.12	1.8	14.4	75.6	91.3
11	-0.37	0.7	9.2	100.4	170.9

**Table 5.1** Data for the histograms of subband 2 to 11: the mean, standard deviation, and mean square error for the fitted probability density functions (mse-values in table should be taken  $\times 10^{-2}$ ).

$$\text{Gaussian distribution} \quad f(x) = \sqrt{\frac{\alpha}{\pi}} e^{-\alpha(x-\mu)^2}, \quad (5.13)$$

$$\text{Laplacian distribution} \quad f(x) = \frac{\alpha}{2} e^{-\alpha|x-\mu|} \quad \text{and} \quad (5.14)$$

$$\text{Cauchy's distribution} \quad f(x) = \frac{\alpha}{\pi} \frac{1}{\alpha^2 + (x - \mu)^2} \quad (5.15)$$

where  $\mu$  is the mean value. Note that the Cauchy probability density function has infinite variance, but it would be used over a finite range of intensity values, which gives a finite variance. The functions were fit to each normalized histogram (*i.e.*, the sum of values equals 1) by adjusting the parameter  $\alpha$  and computing the mean square error of the deviation. The results are given in Table 5.1. The histograms are computed from 100 frames of the sequence “Miss B” that is used in Chapter 6. Although subbands 2 and 3 are best approximated by the Cauchy probability density function, the Laplace function gives a close fit as well. Consequently, a subband could be represented by either a Laplacian or Gaussian probability density function, but neither function suits all subbands. The best fitting function is overlaid on each histogram in Fig. 5.6.

The quantizers could be designed according to the assumed underlying probability density function (so called Max-Lloyd quantizers). The codewords, representing the quantization levels, should be based on the probability of occurrence and could be constructed as a Huffman code. However, subband coding allows adapting the quantization to perceptual criteria, a feature that ought to be utilized. Consequently, the analytic design would be based on the fitted probability density functions but adjusted according to perceptual criteria where such are contrary to the dictates obtained through formal analysis. For example, Gharavi and Tabatabai [GHA86, GHA87, GHA88] suggest the use of quantizers with a “dead zone” for high pass filtered components. This means that the quantizer has a wider zero-level than a quantizer designed analytically, based on the signal’s assumed probability density function. The reason for using a “dead zone” is to remove picture noise, which appears as a low-level signal in the higher bands.

The quantizers of the current implementation have a “dead zone” and symmetric, uniform quantization of the values outside the zero-level. For each of the bands 2 to 11, we adjust only the width of the “dead zone” and the step size. Analogously to the quantizer for band 1, the quantizers have a wide range with variable length codewords for the output levels, which are taken as the mid-point between two quantization levels. The codewords are designed for an exponential distribution and have a limit on the maximal length. This limit constitutes the outer bound of a particular quantizer. Only the nonzero values are transmitted through run-length encoding of their locations. Although this quantizer design is sub-optimal in the sense of mean-square error, it provides a tractable way of designing the ten quantizers. However, the performance, as shown in Chapter 6, is good both perceptually and in coding rate.

### 5.2.3 Run-length encoding

The encoded subbands now have large connected areas of zero-valued samples in each frame. This of course suggests that run-length encoding could be fruitfully employed to indicate the stretches of zero and nonzero valued samples so that only the non-zero values need to be transmitted.

(a) 01100100100100101111111100101001000  
 (b) 01100100100100101111111100101001000  
 (c) 011000000000001111111110011000000

*Figure 5.7 (a) The location information to be run-length encoded is complicated by noise. (b) Spikes—limited by two zero-values on either side—are truncated, and (c) gaps of a single zero-value are bridged.*

Run-length encoding is a technique most easily implemented as a one-dimensional encoding along the rows or columns of a picture. Our findings are intuitively clear: to necessitate the fewest runs, each subband frame should be encoded along the spatial direction in which it has been low pass filtered. Moreover, our study indicates a negligible increase in the number of runs if the encoding is restarted at each row and column, respectively. Encoding is complicated by noise, since isolated values have to be encoded as runs on their own, with a large resultant overhead (see Fig. 5.7 (a)). To eliminate these values, which contribute little to the quality of the reconstructed image, we employ a simple “noise cleaning” by truncating each value with two zero-valued samples on each side in the direction of the run, as shown in Fig. 5.7 (b). A study of sequences of reconstructed images shows no visible degradation due to the truncation of these singular values. It does, however, reduce the number of samples to be encoded. Moreover, short gaps of one zero-valued pixel are bridged to get longer and fewer data runs, illustrated in Fig. 5.7 (c). This decreases the overhead associated with the run length encoding but increases



slightly the amount of data values to be sent. Therefore, only one-pixel-wide gaps are bridged.

The codewords could be any set of fixed or variable length numerals, but the  $B_1$ -codes of Meyr *et alii* [MEY74] are very suitable and their implementation requires only counters. They have also been successfully used by Gharavi and Tabatabai in subband coding of still images [GHA86, GHA87, GHA88].

### 5.3 Complexity and parallelism

The computational complexity of the described subband coding is low: the temporal filters each requires only one addition and one shift operation per value; the length 3 spatial filters require each 2 additions and 2 shifts per value, and the length 5 filters need 5 additions and 3 shifts each. We compare the scheme, as described in Sections 5.2 and 5.3, with two other three-dimensional video coding methods: discrete cosine transform coding with interframe prediction, and intra/interframe DPCM. A comparison to vector quantization has not been included since its complexity cannot be easily compared to that of the above methods. The computations pertain to the analysis, the transform, and the prediction for the respective schemes. Hence, we assume the quantization to be of comparable complexity for all three methods. Although not included, motion compensation and entropy coding can be applied to all the considered schemes.

The complexity has been calculated based on the following criteria:

- i. Subband coding: the implementation shown in Fig. 5.1, with filters given by Eqs. (5.8) to (5.12). A filtering is only performed for the values which are going to be retained in the subsequent subsampling. The number of operations include two additions per pixel for the DPCM encoding performed on band 1.

The complexity of the subband decoder is similar to that of the encoder, since every second sample to be synthesis filtered is zero, whereby the effective filter length is halved.

- ii. Discrete cosine transform with interframe prediction: DCT with blocks of size  $8 \times 8$  and  $16 \times 16$ , where the number of operations is taken from [CHE77]. The transform is followed by interframe prediction, which, per transform coefficient, requires one subtraction for the prediction and one addition for replenishment of the prediction loop. The operations can be either floating point, or the transform vectors can be scaled to a suitable precision of integer arithmetic.
- iii. Intra/interframe DPCM: the complexity is directly proportional to the order of the predictor used. We choose the following predictor:

$$\begin{aligned} \hat{e} = & x[t, i, j] - (a_1 x[t, i - 1, j] + a_2 x[t, i, j - 1] + a_3 x[t, i - 1, j - 1] \\ & + a_4 x[t - 1, i, j] + a_5 x[t - 1, i - 1, j] \\ & + a_6 x[t - 1, i, j - 1] + a_7 x[t - 1, i - 1, j - 1]), \end{aligned} \quad (5.16)$$

where  $\hat{e}$  is the prediction error and  $x[t, i, j]$  is the pixel in frame  $t$ , row  $i$ , and column  $j$ . We consider two cases: the optimal, where all seven coefficients are floating point and possibly different; and a suboptimal case, where  $a_4 = 1$ ,  $a_1 = a_2 = a_5 = a_6 = 1/2$ , and  $a_3 = a_7 = 1/4$ . In addition to the prediction, there is one operation to add the quantized value back to the prediction loop. Depending on the coefficients  $a_i$ , the operations can be floating point, or integer arithmetic, for which some multiplications may be carried out as shift operations.

The number of operations per pixel of the input image is tabulated for the three methods in Table 5.2 [KAR88a]. The subband coding has a complexity which compares favorably to the other schemes. In terms of storage requirement, both

METHOD	ADDS	SHIFTS	MULTS
<i>i</i>	8.9	6.6	0.0
<i>ii</i> 8x8	8.5	0.0	4.0
<i>ii</i> 16x16	11.2	0.0	5.5
<i>iii</i> Opt.	8.0	0.0	7.0
<i>iii</i> Subopt.	8.0	3.0	1.0

**Table 5.2** Operations per pixel for coding methods *i* to *iii*.

the subband coding and the transform/interframe coding require two frames to be stored. The DPCM scheme may operate with one frame and two scan-lines stored; where the frame is stored in the prediction loop and the two scan-lines are the memory needed for the input frame. Apart from the encoding and decoding delays, the subband coder has structural latency of one frame period (33 ms) at the analysis side (none on the synthesis side), which is due to the temporal FIR filtering. Consequently, 2-tap filters may be the maximally tolerable from a delay point of view for duplex video transmissions (see Section 2.4.1). Note that the other methods have a zero latency owing to the recursive structure of their interframe coding. However, some of the latency will be offset by the less complex encoding in the subband case.

In terms of parallel implementation, the structure of the subband coder shown in Fig. 5.1 clearly indicates how the data-flows branch out after each stage of filters, where the rate in each branch has been reduced by a factor 2. The tree structure of Fig. 5.1 may be too complex for a hardware architecture. In that case, the scheme can be made completely parallel by cascading the filters that come in sequence and subsequently subsample in all dimensions at the end. This would yield 11 parallel and independent three dimensional filters, for which the computations can be organized in such a way that there is no increase in arithmetic complexity. The DCT based scheme can also be implemented in parallel since each block of pixels is

processed independently of all others. In contrast to these two schemes, the DPCM is not well-structured for parallel implementation. The prediction gain decreases with each separation of the data. For example, a separation of interlaced video signals to be processed as two separate fields would perform well during periods of high motion in the captured scene, but the prediction would not be the best possible during low motion.

When considering parallel architectures, the computational complexity for each of the parallel branches is of interest. The subband analysis tree in Fig. 5.1 is unbalanced, with the longest branch for band 1. The sequential complexity of that branch is 2.9 additions and 1.9 shifts per pixel in the input, including the DPCM encoding. Consider the coding of a sequence with, say, 30 frames per second of size  $512 \times 480$  pixels, as used in Chapter 6. The subband coding would require maximally 21.4M additions and 14.0M shifts per second.

#### 5.4 Packetization and error concealment

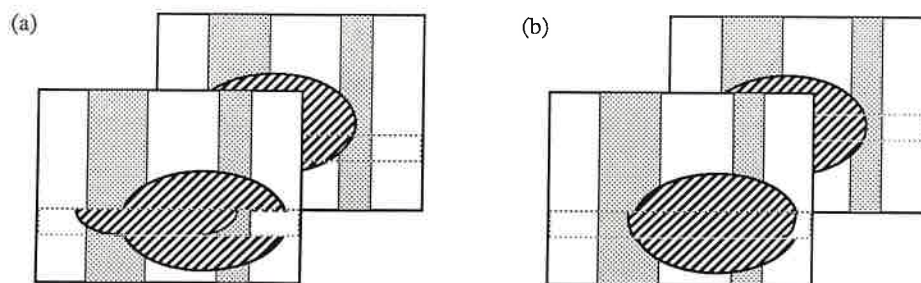
Our implementation includes error concealment through the use of visual redundancy, as discussed in Section 2.3.2; it does not include the use of error correcting codes. Firstly, synchronization flags are inserted into the data stream after the encoding, according to the method of Section 3.2. The synchronization flags are inserted so that neither data runs nor runs of zeroes are broken into different segments. Each individual flag contains the address of the starting location of the first run that follows. Secondly, all subbands are packetized independently of one another. Thus, when a packet is lost, the included data can be treated as erasures, *i.e.*, the locations of the lost data are known. This simplifies the error recovery since there is no error propagation and the erased values may be concealed to alleviate the visibility of the error. Note that although the data from different bands are not

multiplexed into one packet, one packetizer may still serve more than one subband.

Error recovery has been shown to be of importance mainly in the DPCM-encoded band [KAR87]. For the PCM-encoded subbands, the erasures have been found to be masked by the data in other bands, as we show in Chapter 6. A lost packet of PCM values is thus replaced by zero-valued samples (*i.e.*, the mean value). This appears as lost texture and detail in the reconstructed image. The visibility of this distortion is low when viewing is at video rate primarily because the effect is subtle and the erased areas are scattered spatially with varying locations over time. Hence, all further discussion in this section pertains to only the DPCM-encoded subband.

The information contained in the DPCM-encoded subband is the most vital and will be transmitted at a high priority. However, this does not guarantee that packets will not be lost or unduly delayed, even though such loss or delay is extremely rare. Therefore some method of error recovery has to be considered to avoid a complete restart of the video session if a packet is lost and to reduce the visibility of the event. To achieve error recovery of the signal, the signal is segmented as described above with the addition that each synchronization-flag starts with one PCM value and the error signal is packetized along the direction of the prediction. Hereby we have confined the error to the lost values; no further propagation is possible. Note that if we had used two- or three-dimensional prediction for the base-band compression, this erasure property would only have been obtained by including one- or two-dimensional arrays of PCM values, respectively, with each synchronization-flag, which is clearly unfeasible.

In a first attempt to cover the erased area, the values that could not be properly reconstructed were taken from the corresponding area in the previous frame [KAR87]. The difference between the values from the previous frame, which



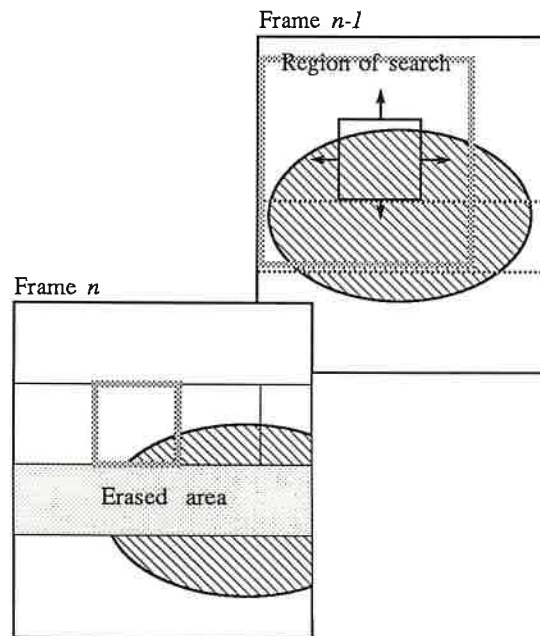
*Figure 5.8 Concealment of an erased image area: (a) the corresponding area of the previous frame has been used, and (b) the best matching area of the previous frame has been used. (The elliptic-shaped area is the moving foreground.)*

have replaced the erasures, and the neighboring correct values was expected to be smoothed out during the subsequent synthesis low pass filtering. This proved to be an unsatisfactory remedy when there were large amounts of motion in the video scene, as illustrated in Fig. 5.8 (a). To improve the performance when there video scene contain motion, we suggest a simple motion compensation scheme to locate the most proper area in the previous frame to be used for the replacement, which is illustrated by Fig. 5.8 (b). We deem the scheme possible to be implemented with low complexity. As justification consider that the error recovery has to be invoked only when a packet is lost; that the frame rate is only half the input rate; and that the motion estimation may be done only for a small number of pixels, for example, the corner pixels of a rectangular area circumscribing the erased area. The method has been given a preliminary investigation as described below.\*

Assume that the signal is packetized along the scan-lines. A lost packet will hence correspond to one or more lines missing. Once the location and extent of the loss has been determined, displacement estimates are computed for strips of blocks, one above and one below the lost area, by using a standard block-matching technique (see for example [MUS85]). It is computationally intractable to match

---

\* The work was done by Kamil Metin Uz.



*Figure 5.9 Estimation of local displacement. Given the marked block in frame  $n$ , a restricted region is searched in the previous frame,  $n - 1$ , for the location of the block which most closely matches the given one. The block size is typically in the range of  $5 \times 5$  up to  $15 \times 15$ .*

the block by an exhaustive search, therefore a heuristic technique is applied. A maximum displacement is established based on temporal sampling rate and picture characteristics. This defines a region to be searched in the previous frame. Next, a distance measure for the matching must be defined. This is taken as the sum of absolute differences of the original block and the displaced block in the previous frame and it is the measure to be minimized. Given these parameters, the target frame can be searched for each of the blocks surrounding the erased area, as illustrated in Fig. 5.9. The search goes on until a local minimum of the error function is reached. A problem arises when the lost area from the previous frame is to be replaced; the displacement estimates, computed above and under the lost area, do not necessarily agree. A good compromise was found to be median filtering of the estimates. The choice of block size depends on two parameters. The first is the height of the lost area, which should be comparable in height to block size, and

the second parameter is the smallest feature that must be recovered, which must be larger than the block size. The results are reported in Chapter 6 and they are encouraging.

### 5.5 Extensions to the coding scheme

As pointed out in the introduction to this chapter, the described coding scheme could be augmented by non-separable filter banks. For example, regular interlaced video has a quincux vertical-temporal subsampling structure (*i.e.*, the axes of the sampling lattice are rotated  $45^\circ$  with respect to the temporal and vertical axes). A non-separable filter bank could perform a subband division which better separates high temporal frequencies from high vertical frequencies. However, well-performing non-separable filter banks have to be designed, which is deemed to be outside the scope of the thesis. The non-separable theory appears to find wide application in signal processing of the future high-definition television for which signal formats based on hexagonal and quincux sampling are being discussed (see for example [SCH88, ANS88]). It is worth noting in this context that subband analysis may also be used for size reduction if upsampling and synthesis filtering is by-passed [LEG88a], which may help in resolving problems of downward compatibility from HDTV format to regular TV format.

If robustness is less of an issue, the encoding may be more sophisticated than the outlined method of DPCM/PCM combined with run-length coding. A substantial gain could be achieved by incorporating motion compensation into the system (see for example [MUS85]). If a block matching technique is used, the temporal filtering, with the filters of Eq. (5.8), would be performed by

$$y_l(t) = \frac{1}{2}(x_i(t) + \hat{x}_i(t-1)) \quad \text{and} \quad y_h(t) = \frac{1}{2}(x_i(t) - \hat{x}_i(t-1)). \quad (5.17)$$



We have denoted the  $i$ th block in frame  $t$  by  $x_i(t)$  and the matching block in the previous video frame by  $\hat{x}_i(t-1)$ . The temporal low- and high-pass components,  $y_l(t)$  and  $y_h(t)$ , would be then be temporally subsampled and analyzed spatially, as before. The system still yields perfect reconstruction as long as the receiver gets the motion vectors uncorrupted by the transmission. Recall that due to delay constraints, the length of the temporal filters is restricted to only a few taps (each tap, apart from the first, adds 33 ms delay). The use of recursive temporal filters would therefore be natural in order to improve the subband separation without increasing the delay. Such filters may also be motion compensated for enhanced performance [KRO88].

Quad tree coding has been found to effectively compress loosely correlated images [STR88]. Hence it could be employed instead of run-length coding to compress the subbands. Also, the PCM-quantizations could be adapted to the intensity level of subband 1, which represent the local mean of an image. Thereby the quantization could be performed according to Weber's law: the quantization error becomes higher in bright areas than in low-intensity areas. The PCM encoding of subbands would benefit from further research into bit-allocation procedures for video signal, as reported in [RAM86, WES88b, WES88c]. The subband encoding may of course be changed to any of the more powerful methods tried for image compression: DPCM for all bands [WOO86], vector quantization [WES87, WES88a], or DCT [LEG88a]. In short, the multitude of augmentations to the coding scheme make us certain that subband coding can be successfully applied to all video formats, from HDTV down to video-telephony.

Finally, note that color may be easily incorporated by going from luminance, solely, to a YIQ or YUV color space. The chrominance components are encoded the same

way as the intensity component, but they are reduced to half its spatial dimensions. For a general overview of compression of color images see [LIM77] and for details on subband color coding [GHA87, GHA88]. After encoding, the chrominance information and the intensity information of a sub-signal in a hierarchical coding scheme should preferably be multiplexed after compression so that they experience the same transmission conditions.

## 5.6 Summary

We have investigated a video coding scheme based on the technique of subband coding. Our method relies on three-dimensional analysis of the video which yields eleven subbands with various combinations of high and low temporal and spatial frequencies. The separable filter bank is non-paraunitary but yields perfect reconstruction when there is neither coding nor transmission loss. Of the eleven bands, one is DPCM encoded and ten are encoded by PCM. The quantizers have a wide zero-level, which removes low-amplitude picture noise, and uniform symmetric levels for the non-zero parts. The quantization levels are represented by variable length codewords. Only the non-zero values of the quantized prediction error and the PCM-encoding are transmitted and their locations are given by a run-length code. The coding scheme has an architectural simplicity suitable for parallel implementation and provides a tractable integration with networking issues. Furthermore, its implementation is multiplication free; the only operations needed are integer additions and shifts. The coding scheme was augmented by an error recovery method based on concealment. For ten of the eleven subbands missing values are replaced by mean-valued samples. For the first subband, the corresponding area of the previous video frame was used to replace a missing area. It was discussed how this could be improved by using motion estimation to find

the most appropriate area for replacement. Extensions to the coding scheme were described relating to non-separable subsampling and filter banks, more complex subband encoding, incorporation of motion compensation into the subband analysis, and the coding of color sequences. Results from simulations of the coding scheme and the associated error recovery procedure are presented in Chapter 6.