Chair of Multimedia Communications and Signal Processing
Friedrich-Alexander University Erlangen-Nürnberg

Image and Video Compression

Prof. Dr.-Ing. A. Kaup
M.Sc. F. Brand

Exercises

Summer Term 2021

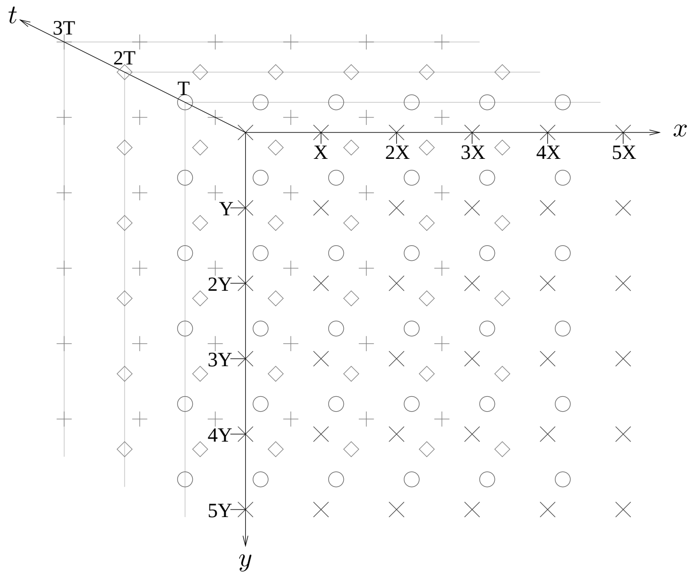
FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Part I
Tasks

1 2D Sampling

For conventional TV usually the interlaced mode is used.

a) Highlight those pixels in the following chart which are rendered and shown in interlaced mode. The first picture is marked with crosses, the second with circles, the third with diamonds and the fourth with plus signs.



b) For explaining the interlaced mode, properties of the Quincunx-Raster are used. How does this correlate with the recently drawed diagram in (a)?

c) Draw the Qunicunx-Raster used by the interlaced mode in frequency domain. Label the axes according to (a).

An LCD-monitor with PAL resolution has 720 pixels in x- direction and 576 pixels in y- direction. Now, a sinusoidal oscillation in interlaced imaging with 8 pixels per oscillation is displayed in y- direction.

d) This oscillation is now moved in y- direction. For how many pixels per time step may this oscillation maximally be moved such that no aliasing occurs?

e) Which visual effect is caused by the aliasing, if the oscillation is moved faster?

f) If the oscillation is now displayed and moving in x- direction, is the maximum movement speed for alias free rendering increased or decreased?

2 Entropy Coding

2.1 Unary and Golomb Codes

We consider a memoryless source with an alphabet of all non-negative integers $A_x = \{0, 1, 2, 3, ...\}$ and a geometric distribution $p_x(x) = 2^{-(x+1)}$.

a) Construct the codewords for the first six symbols of A_x using Unary Coding. Use '1' to terminate a codeword.

b) Show that the unary code is an optimum code for the given source.

c) The symbols now occur according to the following probability mass function $p_x(x) = 0.3 \cdot 0.7^x$. Estimate the Golomb codewords for the first six symbols of A_x, while choosing the integer divisor $m = 2$.

d) Decode the code 010100010100 using a Golomb Code, $m = 16$.

2.2 Huffman Coding

Now alphabet $B_x = \{1, 2, 3\}$ is given. The probabilities of the symbols are given in Table 1.

symbol	probability
1	0.50
2	0.41
3	0.09

Table 1: Symbol probabilities

a) Calculate the mean and variance of the output signal.

b) Calculate the self information for each symbol and the entropy of the signal.

c) Create a Huffman code for this signal and code the symbols 1-2-3-1-2-1-2-1. Assign always '0' to the upper branch.

d) What is the lower and upper bound for the average code word length, when coding this input signal using a Huffman code? Determine the average code word length of the generated Huffman code and compare it to the entropy. Calculate the coding redundancy.

Now a Vector Huffman Code is considered. Therefore combine 2 symbols into 2-dimensional hyper-symbols

e) Create a Vector Huffman code for the hyper-symbols and code the symbols 1-2-3-1-2-1-2-1. Assign always '0' to the upper branch.

f) For the generated Huffman code in (e), calculate the redundancy per hyper-symbol and the redundancy per symbol.

2.3 Arithmetic Coding

Now alphabet $C_x = \{0, 1, 2, 4, 7, 9\}$ is given. The probabilities of the symbols are given in Table 2.

symbol	probability
0	0.50
2	0.11
4	0.03
6	0.15
8	0.04
9	0.17

Table 2: Symbol probabilities

a) An arithmetic coder shall be used to transmit the sequence 6-0-9-0. We assume, that the decoder knows, that always 4 symbols are transmitted together, so we do not need an EOF symbol. Calculate the expected code word length for this example.

b) Code the symbols of the sequence 6-0-9-0. Start the interval for the grey value 0 at the lower end (at 0) of the coding interval and count up by 1.

c) Decode the following transmission using the arithmetic code: 1000011111

3 Quantization and Entropy Coding

3.1 Quantization

We have an image source which has a triangular probability mass function. It generates a sample image that looks like this:

$$\boldsymbol{Y} = \begin{bmatrix} -2 & 1 & -1 & 0 & -3 \\ 9 & -1 & 0 & -7 & 1 \\ 7 & 1 & 3 & -9 & -1 \end{bmatrix}$$

a) Draw the function of a midrise and a midtread quantizer with a stepsize of 3. Quantize the input image using both quantizers.

b) We now assume that the values near zero contain no significant information but are generated by noise. Which of the quantizers is better suited then? Can the performance be improved by using another quantizer type?

In the following we assume, that all amplitude values are equally important. Therefore the quantization shall retain all values with the same error level, that is: the quantization error variance shall be minimized.
The following discrete triangular probability mass function of the amplitude values is provided:

$p_x(0)$	$p_x(1) = p_x(-1)$	$p_x(2) = p_x(-2)$	$p_x(3) = p_x(-3)$	$p_x(4) = p_x(-4)$	$p_x(5) = p_x(-5)$	$p_x(6) = p_x(-6)$	$p_x(7) = p_x(-7)$	$p_x(8) = p_x(-8)$	$p_x(9) = p_x(-9)$	$p_x(10) = p_x(-10)$
$\frac{10}{100}$	$\frac{9}{100}$	$\frac{8}{100}$	$\frac{7}{100}$	$\frac{6}{100}$	$\frac{5}{100}$	$\frac{4}{100}$	$\frac{3}{100}$	$\frac{2}{100}$	$\frac{1}{100}$	$\frac{0}{100}$

c) Calculate two iterations of the Lloyd-Max algorithm. Start with the midtread quantizer with the stepsize 3 and 7 reconstruction values.

d) Calculate the variance of the quantization error for the old uniform midtread quantizer and the new optimized quantizer. Use the discrete probabilities given at the start of this exercise and explain the result.

3.2 Quantization followed by Entropy Coding

a) Now, we will use the uniform midtread quantizer and do a run-level and huffman coding. Write down the indices for the reconstruction levels for the image using numbers from -3 to 3. Use line-by-line scanning and write down all the run and level combinations. Use the following table to obtain the number of bits used.

Run	Levelindex	Nr. of Bits	Bit pattern
0	1	3	10s
0	2	5	1111 s
0	3	7	0101 01s
0	4	8	0010 111s
0	5	9	0001 1111 s
0	6	10	0001 0010 1s
0	7	10	0001 0010 0s
0	8	11	0000 1000 01s
0	9	11	0000 1000 00s
0	10	12	0000 0000 111s
0	11	12	0000 0000 110s
0	12	12	0000 0100 000s
1	1	4	110s

Run	Levelindex	Nr. of Bits	Bit pattern
1	2	7	0101 00s
1	3	9	0001 1110 s
1	4	11	0000 0011 11s
1	5	12	0000 0100 001s
1	6	13	0000 0101 0000 s
2	1	5	1110 s
2	2	9	0001 1101 s
2	3	11	0000 0011 10s
2	4	13	0000 0101 0001 s
3	1	6	0110 1s
3	2	10	0001 0001 1s
3	3	11	0000 0011 01s
...			

Table 3: VLC coding table of H.263 [ITU-T Rec. H.263 (01/2005): Table 16 on Page 41], the "s" in the table denotes the sign bit, 1 for negative and 0 for positive quantization indices

b) Create a Huffman code for the uniform midtread quantizer and code the image. How many Bits are necessary? Compare the result with the run-level-Huffman coding.

3.3 Vector Quantization

A vector quantizer shall be used to encode two-dimensional vectors $\mathbf{x} = (x_1, x_2)^T$. Thus, a codebook has to be calculated using the LBG algorithm. The following training vectors $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{12}$ shall be used:

$$\mathbf{x}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \mathbf{x}_2 = \begin{pmatrix} 1 \\ 4 \end{pmatrix} \quad \mathbf{x}_3 = \begin{pmatrix} 3 \\ 2 \end{pmatrix} \quad \mathbf{x}_4 = \begin{pmatrix} -1 \\ 2 \end{pmatrix} \quad \mathbf{x}_5 = \begin{pmatrix} -2 \\ -2 \end{pmatrix} \quad \mathbf{x}_6 = \begin{pmatrix} -3 \\ 1 \end{pmatrix}$$

$$\mathbf{x}_7 = \begin{pmatrix} 3 \\ 3 \end{pmatrix} \quad \mathbf{x}_8 = \begin{pmatrix} 5 \\ -2 \end{pmatrix} \quad \mathbf{x}_9 = \begin{pmatrix} 2 \\ -5 \end{pmatrix} \quad \mathbf{x}_{10} = \begin{pmatrix} -5 \\ -4 \end{pmatrix} \quad \mathbf{x}_{11} = \begin{pmatrix} -2 \\ -4 \end{pmatrix} \quad \mathbf{x}_{12} = \begin{pmatrix} -5 \\ -2 \end{pmatrix}$$

The initial set of reconstruction vectors (or representatives) $\hat{\mathbf{x}}_q$ consists of the following $M = 4$ vectors:

$$\hat{\mathbf{x}}_0^{(0)} = \begin{pmatrix} 3 \\ 3 \end{pmatrix} \quad \hat{\mathbf{x}}_1^{(0)} = \begin{pmatrix} -3 \\ 3 \end{pmatrix} \quad \hat{\mathbf{x}}_2^{(0)} = \begin{pmatrix} -3 \\ -3 \end{pmatrix} \quad \hat{\mathbf{x}}_3^{(0)} = \begin{pmatrix} 3 \\ -3 \end{pmatrix}$$

a) Draw the partitions of the vector quantizer using the initial reconstruction vectors.

b) Calculate the first iteration of the LBG algorithm. Write down the new sets of training vectors B_q and the new reconstruction vectors $\hat{\mathbf{x}}_q^{(1)}$.

c) Draw the partitions of the vector quantizer after the first iteration of the LBG algorithm.

4 Linear Prediction and DPCM

The following signal shall be analyzed and transmitted:

$$x[n] = [8, 2, -4, -8, -16, -20, 0, 0, 0, 0, 0, 4, 8, 12, 16] \quad n = 0, 1, \ldots, 14$$

We assume, that the signal has a mean value of 0.

a) Draw the block diagram of a predictive Coder and Decoder.

b) Draw a block diagram of a linear predictor, which uses the last 3 values for prediction. Assume, that the signal may be modeled using a AR(3) process.

c) Write down the Wiener-Hopf equation to compute the weights for the predictor from the previous task **b)**.

d) Estimate the first two elements of the matrix from the signal given in this exercise. Limit the length of the signal where necessary, e.g., when the signal values at $x[n]$ and $x[n-1]$ are needed, the length of the signal is reduced by 1.

The inverse matrix ${\boldsymbol{\Phi}_{\boldsymbol{xx}}}^{-1}$ looks like this:

$${\boldsymbol{\Phi}_{\boldsymbol{xx}}}^{-1} = \begin{bmatrix} 0.0229 & -0.0193 & 0.0058 \\ -0.0193 & 0.0378 & -0.0193 \\ 0.0058 & -0.0193 & 0.0229 \end{bmatrix}$$

e) Calculate the prediction coefficients a_1, a_2 and a_3.

f) Assume that the signal was 0 from its start up to the time 0. Calculate the first 5 values of the error signal $e[n]$.

g) Decode the signal again.

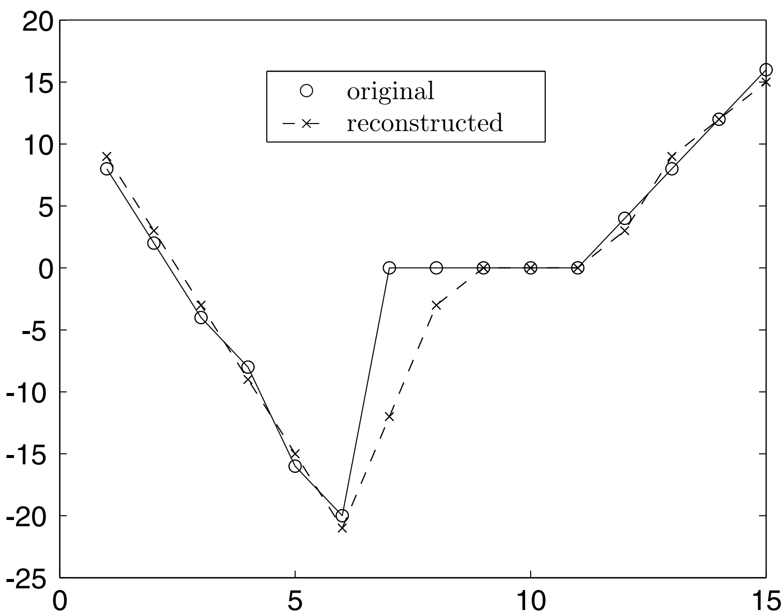
h) Can you start the decoding process at an arbitrary position?

i) The error signal $e[n]$ shall now be quantized before transmission. Specify an algorithm to do this without encoder-decoder mismatch. Sketch a block diagram for this algorithm.

j) Mark the decoder that is hidden in the encoder.

k) The predictor will now be simplified. It only contains one coefficient, $a_1 = 1$. The quantizer shall be a uniform midtread quantizer with stepsize 3 and 7 reconstruction values. Calculate the error signal $e'[n]$ and the signal that will be decoded $x'[n]$.

l) In this plot the resulting signal and the original one can be seen. Explain where you can see which artifacts.



5 Transforms

5.1 General questions about transforms

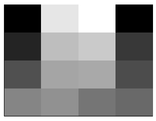
a) What are the advantages of the DCT when compared to the DFT?

b) Sketch the computation of the Karhunen-Loéve Transform. Describe the advantages and disadvantages of this transform.

5.2 Separable Transform

To transmit the following picture, it has to be transformed and quantized:

$$\boldsymbol{X} = \begin{bmatrix} 11 & 212 & 233 & 11 \\ 44 & 175 & 186 & 62 \\ 83 & 154 & 158 & 77 \\ 127 & 136 & 114 & 102 \end{bmatrix}$$



The following transform matrix shall be used. It is a separable transform.

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}$$

a) Using this transform matrix, how is the transformed signal $\boldsymbol{Y}$ generated?

b) What is the main advantage of a separable transform?

c) Is this transform orthogonal?

d) Is the transform orthonormal? In case it is not orthonormal, can we change it into an orthonormal one?

In the following we use a modified transform matrix:

$$\boldsymbol{A} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{2}{\sqrt{10}} & \frac{1}{\sqrt{10}} & -\frac{1}{\sqrt{10}} & -\frac{2}{\sqrt{10}} \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\ \frac{1}{\sqrt{10}} & -\frac{2}{\sqrt{10}} & \frac{2}{\sqrt{10}} & -\frac{1}{\sqrt{10}} \end{bmatrix}$$

e) The input signal shall now be transformed. Fill in the missing values.

$$\boldsymbol{Y}_C = \boldsymbol{A} \cdot \boldsymbol{X} = \begin{bmatrix} 132.5 & \boxed{} & 345.5 & 126 \\ -85.6977 & 54.7074 & 84.1166 & -62.2969 \\ 5.5 & \boxed{} & 1.5 & -13 \\ -12.0167 & 10.7517 & 19.9223 & -19.2899 \end{bmatrix}$$

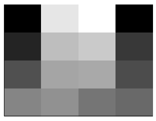
$$\boldsymbol{Y} = \boldsymbol{Y}_C \cdot \mathbf{A}^T = \begin{bmatrix} 471.25 & 1.89737 & \boxed{} & 6.48267 \\ -4.5853 & -24.1 & -143.409 & 11.2 \\ 1.75 & 14.2302 & -9.25 & 0.790569 \\ -0.316228 & 1.7 & -30.9903 & 8.1 \end{bmatrix}$$

f) Apply the following quantizer table to the transformed signal $\boldsymbol{Y}$. Specify both the quantizer indices (levels) and the quantized signal. The quantizer itself is a midtread quantizer without dead-zone.

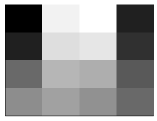
$$\boldsymbol{Q} = \begin{bmatrix} 4 & 8 & 16 & 32 \\ 8 & 16 & 32 & 64 \\ 16 & 32 & 64 & 64 \\ 32 & 64 & 64 & 64 \end{bmatrix}$$

The following plot shows the original and the inverse transformed signal (rounded to integers):

$$\boldsymbol{X} = \begin{bmatrix} 11 & 212 & 233 & 11 \\ 44 & 175 & 186 & 62 \\ 83 & 154 & 158 & 77 \\ 127 & 136 & 114 & 102 \end{bmatrix}$$



$$\boldsymbol{X}_Q = \begin{bmatrix} 10 & 202 & 214 & 36 \\ 38 & 186 & 192 & 51 \\ 94 & 154 & 148 & 81 \\ 122 & 138 & 126 & 96 \end{bmatrix}$$



g) What is the equation for the inverse transformation to get the signal $\boldsymbol{X}_Q$?

5.3 Discrete Cosine Transform

A digital image is divided in matrices of the size 4x4 pixels. The following two matrices are given:

$$\text{matrix 1: } \boldsymbol{M}_1 = \begin{bmatrix} 100 & 100 & 100 & 100 \\ 100 & 100 & 100 & 100 \\ 100 & 100 & 100 & 100 \\ 100 & 100 & 100 & 100 \end{bmatrix}$$

$$\text{matrix 2: } \boldsymbol{M}_2 = \begin{bmatrix} 0 & 100 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 100 & 0 & 0 \end{bmatrix}$$

The matrices are transformed by using the discrete cosine transform.

$$\text{transformed matrix 1: } \boldsymbol{M}_{1,T} = \begin{bmatrix} \boxed{} & 0 & 0 & 0 \\ 0 & \boxed{} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\text{transformed matrix 2: } \boldsymbol{M}_{2,T} = \begin{bmatrix} 100 & \boxed{} & -100 & \boxed{} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

a) Calculate the missing transform coefficients.

The transformed matrices are quantized using the following matrix:

$$\boldsymbol{Q} = \begin{bmatrix} 9 & 15 & 20 & 30 \\ 15 & 12 & 25 & 35 \\ 20 & 25 & 30 & 40 \\ 30 & 35 & 40 & 50 \end{bmatrix}$$

b) Specify the quantizer indices (levels).

c) Calculate the quantized matrices.

Inverse transform:

$$\text{inverse transformed matrix 1: } \boldsymbol{M}_{1,T,Q,T^{-1}} = \begin{bmatrix} \square & \square & \square & \square \\ \square & \square & \square & \square \\ \square & \square & \square & \square \\ \square & \square & \square & \square \end{bmatrix}$$

$$\text{inverse transformed matrix 2: } \boldsymbol{M}_{2,T,Q,T^{-1}} = \begin{bmatrix} \square & \square & \square & \square \\ \square & \square & \square & \square \\ \square & \square & 2,43 & \square \\ \square & \square & \square & \square \end{bmatrix}$$

d) Fill in the missing values.

e) Estimate the error of the reconstructed matrices.

6 Subband-Coding

6.1 General Properties

a) Draw the block diagram for a subband coder and decoder using octave band splitting such that 3 subbands are obtained.

6.2 Subband decomposition

The following signal $x[k]$ shall be transmitted using a subband coder that includes the symmetric bi-orthogonal filters $H_0(z)$ and $H_1(z)$.

$$x[k] = \begin{bmatrix} -20 & -50 & -30 & 50 & 30 & -60 & 60 & -60 & 60 & 10 & 5 & 10 & 5 & 5 \end{bmatrix}$$

The z-transform of the lowpass and the z-transform of the highpass filter are given:
Lowpass:

$$H_0(z) = \frac{1}{8}\left(-z^1 + 2 + 6z^{-1} + 2z^{-2} - z^{-3}\right)$$

Highpass:

$$H_1(z) = \frac{1}{2}\left(-z^1 + 2 - z^{-1}\right)$$

a) We assume, that the signal is 0 outside the specified range. Calculate the first four signal values after the first low and highpass filters.

b) How can the synthesis filters be derived from the analysis filters? Calculate the coefficients of the synthesis filters.

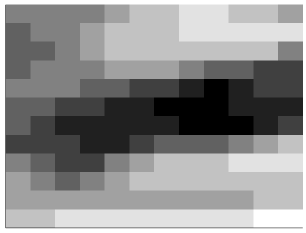
c) In which standard are these filters used?

7 Motion Estimation and Color Spaces

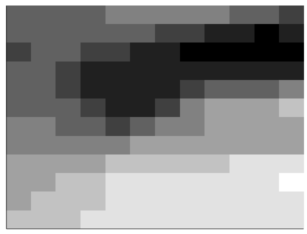
7.1 Motion Estimation

We want to predict a picture from its temporal predecessor. To achieve this, we want to use a block-based motion compensation. The two pictures look as follows:

$$\boldsymbol{Y}_0 = \begin{array}{cc} & \xrightarrow{m} \\ n \downarrow & \begin{bmatrix} 230 & 236 & 237 & 246 & 274 & 295 & 311 & 329 & 329 & 305 & 286 & 265 & 240 & 207 & 180 & 166 \\ 218 & 230 & 246 & 270 & 286 & 292 & 308 & 323 & 335 & 341 & 343 & 328 & 300 & 246 & 201 & 177 \\ 206 & 221 & 237 & 267 & 287 & 296 & 306 & 306 & 309 & 309 & 290 & 254 & 233 & 206 & 186 & 174 \\ 212 & 233 & 244 & 253 & 271 & 274 & 262 & 244 & 216 & 198 & 191 & 194 & 230 & 248 & 247 & 228 \\ 226 & 238 & 236 & 222 & 205 & 193 & 172 & 148 & 133 & 163 & 187 & 171 & 193 & 222 & 238 & 259 \\ 223 & 199 & 176 & 167 & 156 & 147 & 129 & 114 & 117 & 141 & 156 & 147 & 183 & 245 & 231 & 207 \\ 200 & 167 & 164 & 161 & 153 & 150 & 140 & 131 & 126 & 129 & 151 & 181 & 236 & 311 & 290 & 232 \\ 188 & 176 & 170 & 155 & 155 & 182 & 200 & 203 & 211 & 241 & 278 & 288 & 301 & 319 & 309 & 312 \\ 227 & 209 & 178 & 184 & 229 & 272 & 293 & 302 & 311 & 323 & 331 & 329 & 324 & 312 & 307 & 319 \\ 268 & 238 & 212 & 233 & 276 & 291 & 303 & 309 & 309 & 309 & 295 & 286 & 296 & 302 & 313 & 325 \\ 267 & 258 & 264 & 264 & 262 & 259 & 265 & 271 & 275 & 275 & 285 & 300 & 319 & 331 & 334 & 337 \\ 296 & 308 & 322 & 322 & 316 & 319 & 326 & 335 & 337 & 337 & 350 & 362 & 364 & 358 & 353 & 350 \\ 316 & 328 & 338 & 347 & 355 & 364 & 370 & 373 & 381 & 378 & 377 & 374 & 359 & 359 & 362 & 362 \\ 319 & 331 & 344 & 347 & 352 & 358 & 358 & 361 & 366 & 369 & 371 & 368 & 353 & 362 & 372 & 365 \\ 328 & 340 & 347 & 350 & 356 & 356 & 357 & 357 & 360 & 363 & 364 & 367 & 364 & 370 & 381 & 375 \\ 331 & 340 & 347 & 350 & 356 & 359 & 357 & 354 & 354 & 360 & 364 & 370 & 373 & 370 & 372 & 369 \end{bmatrix} \end{array}$$



$$\boldsymbol{Y}_1 = \begin{bmatrix} 227 & 227 & 227 & 243 & 257 & 266 & 278 & 278 & 266 & 245 & 226 & 214 & 221 & 203 & 206 & 224 \\ 227 & 230 & 239 & 248 & 241 & 232 & 222 & 204 & 189 & 165 & 152 & 167 & 206 & 188 & 191 & 236 \\ 224 & 233 & 232 & 218 & 195 & 189 & 180 & 162 & 153 & 138 & 140 & 164 & 184 & 176 & 200 & 262 \\ 230 & 236 & 215 & 182 & 174 & 171 & 168 & 165 & 171 & 168 & 176 & 185 & 218 & 243 & 276 & 311 \\ 235 & 229 & 209 & 188 & 177 & 168 & 175 & 211 & 232 & 241 & 250 & 262 & 290 & 305 & 311 & 317 \\ 248 & 238 & 234 & 219 & 186 & 189 & 224 & 266 & 292 & 298 & 306 & 321 & 327 & 318 & 306 & 303 \\ 261 & 258 & 253 & 241 & 222 & 237 & 265 & 280 & 292 & 298 & 301 & 307 & 309 & 306 & 304 & 313 \\ 278 & 281 & 279 & 273 & 276 & 288 & 295 & 301 & 305 & 308 & 310 & 307 & 313 & 328 & 338 & 344 \\ 293 & 299 & 302 & 308 & 328 & 334 & 333 & 342 & 343 & 349 & 350 & 350 & 352 & 361 & 367 & 367 \\ 299 & 311 & 317 & 329 & 345 & 354 & 356 & 359 & 364 & 373 & 371 & 377 & 376 & 373 & 367 & 364 \\ 303 & 315 & 324 & 336 & 348 & 351 & 354 & 360 & 363 & 372 & 370 & 364 & 365 & 368 & 372 & 369 \\ 315 & 318 & 330 & 345 & 351 & 354 & 360 & 363 & 366 & 366 & 364 & 364 & 365 & 365 & 375 & 378 \\ 318 & 330 & 343 & 349 & 350 & 362 & 368 & 368 & 368 & 365 & 366 & 372 & 371 & 371 & 371 & 374 \\ 321 & 336 & 343 & 349 & 356 & 359 & 365 & 368 & 368 & 365 & 369 & 372 & 374 & 380 & 383 & 377 \\ 317 & 332 & 332 & 347 & 353 & 356 & 356 & 356 & 362 & 362 & 366 & 369 & 376 & 379 & 391 & 388 \\ 317 & 329 & 335 & 344 & 353 & 356 & 353 & 356 & 359 & 359 & 369 & 372 & 382 & 382 & 385 & 382 \end{bmatrix}$$



The motion estimation shall be done on a blocksize of $M \times N = 2 \times 2$. The measure for the difference is the sum of absolute differences:

$$\mathrm{SAD}(d_m, d_n) = \sum_{m=x}^{x+M-1} \sum_{n=y}^{y+N-1} \left| Y_1(m,n) - Y_0(m+d_m, n+d_n) \right|$$

where (x,y) is the position of the upper left pixel of the block.

a) Divide the picture that shall be analyzed into blocks of size 2×2. Now we look at the block in the fourth row and column (counting from 1). Calculate the SAD for the motion vectors of $(d_m, d_n) = (0, 0)$ and $(d_m, d_n) = (-1, 0)$, that is: the current position and one pixel left of it. Which position would be better?

b) For the block we are looking at, all the SADs of all possible motion vectors were calculated. The SAD for the motion vector $(0, 0)$ is marked. Do a three-step search to get a suitable motion vector. Is the result optimal?

$$\mathrm{SAD} = \begin{bmatrix}
227 & 192 & 142 & 65 & 42 & 71 & 130 & 175 & 169 & 134 & 111 & 74 & 158 & 307 & 417 \\
266 & 207 & 121 & 53 & 46 & 61 & 102 & 132 & 153 & 164 & 178 & 192 & 226 & 302 & 403 \\
269 & 206 & 140 & 81 & 89 & 117 & 157 & 206 & 255 & 261 & 262 & 230 & 224 & 254 & 306 \\
232 & 190 & 186 & 190 & 210 & 258 & 315 & 400 & 431 & 402 & 398 & 353 & 248 & 186 & 169 \\
255 & 292 & 340 & 391 & 440 & 500 & 578 & 629 & 587 & 494 & 480 & 447 & 298 & 205 & 206 \\
352 & 435 & 473 & 504 & 535 & 575 & 627 & 653 & 628 & 564 & 506 & 394 & 186 & 96 & 181 \\
410 & 464 & 491 & 517 & 501 & 469 & \boxed{467} & 470 & 434 & 342 & 243 & 135 & 84 & 88 & 98 \\
341 & 408 & 454 & 418 & 303 & 194 & 145 & 148 & 131 & 84 & 85 & 101 & 115 & 106 & 106 \\
199 & 304 & 334 & 219 & 73 & 26 & 66 & 90 & 111 & 129 & 130 & 122 & 93 & 93 & 123 \\
116 & 169 & 168 & 106 & 97 & 121 & 127 & 123 & 119 & 95 & 47 & 60 & 107 & 139 & 168 \\
32 & 57 & 65 & 61 & 63 & 70 & 74 & 87 & 93 & 106 & 156 & 204 & 231 & 235 & 233 \\
107 & 155 & 188 & 199 & 213 & 238 & 263 & 285 & 292 & 301 & 322 & 318 & 299 & 291 & 286 \\
153 & 200 & 235 & 260 & 288 & 309 & 321 & 340 & 353 & 354 & 349 & 313 & 292 & 314 & 320 \\
177 & 221 & 247 & 264 & 281 & 288 & 292 & 303 & 317 & 326 & 329 & 311 & 308 & 344 & 352 \\
198 & 233 & 253 & 271 & 286 & 288 & 284 & 284 & 296 & 310 & 324 & 333 & 336 & 352 & 356
\end{bmatrix}$$

7.2 Color Spaces

In a video coder, YCbCr color coordinates are used for representing the color values. Due to a programming error, Cb and Cr are mixed up.

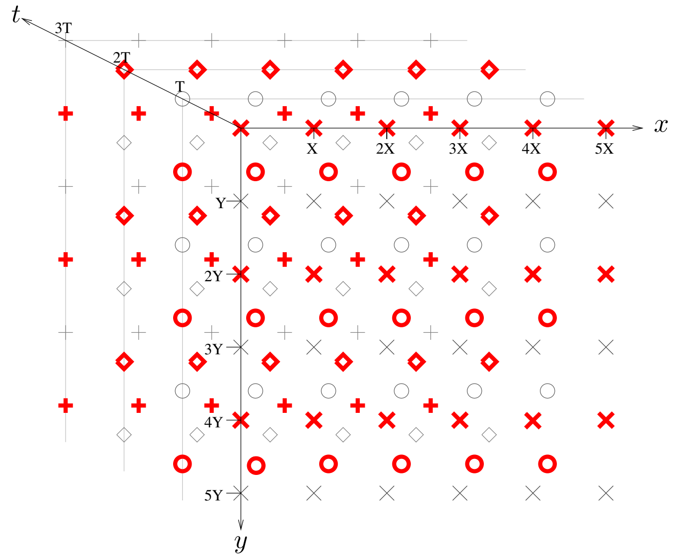
a) Calculate the received and decoded RGB signal for [r g b]=[255 255 0].

b) Which Color is represented?

Part II
Proposed Solutions

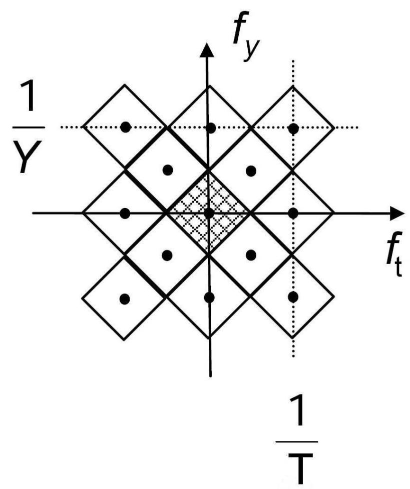
1 2D Sampling

a) Highlight those pixels in the following chart, which are rendered and showed in interlaced mode. The first picture is marked with crosses, the second with circles, the third with rhombuses and the fourth with plus signs.



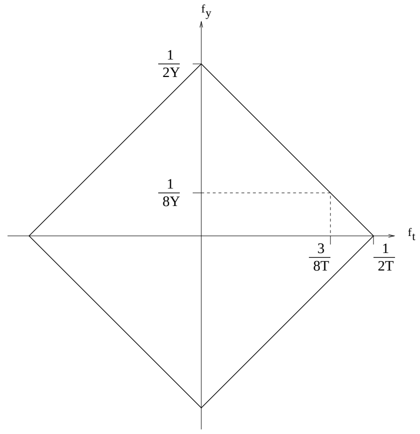
b) The Quincunx-Raster is applied in y-t direction.

c) Draw the Qunicunx-Raster depending on interlaced mode in frequency domain. Label the axes according to (a).



d) Solution in frequency domain:

oscillation period: $8Y \rightarrow$ oscillation frequency: $\frac{1}{8Y}$



max. temporal frequency: $\frac{3}{8T} \rightarrow$ temporal oscillation period: $\frac{8T}{3}$
If the earliest time of displaying the sinusoidal oscillation identically occurs after $\frac{8}{3}T$, it has to move with less than 8 pixel $\cdot \frac{3}{8T} = 3$ per timestep. Thus at 3 pixels per timestep aliasing already occurs.

e) Visual effects:

- movement of the sinusoidal oscillation changes direction
- magnitude of movement changes

f) Movement speed can be increased, because the sampling grid in x direction has a higher cut-off-frequency regarding the absolute resolution in oscillations per degree.

2 Entropy Coding

2.1 Unary and Golomb Codes

a) Unary codewords:

symbol x	Unary codeword c_x	length L_x
0	1	1
1	01	2
2	001	3
3	0001	4
4	00001	5
5	000001	6

Table 4: Unary Coding

b) Optimum code: $r = R - H(X) = 0$

Proof: $R = H(X)$

$$R = \sum_{x \in A_x} L_x \cdot p_x(x) = \sum_{x \in A_x} (x+1) \cdot p_x(x)$$

$$\begin{aligned} H(X) = & \quad - \textstyle\sum_{x \in A_x} \log_2(p_x(x)) \cdot p_x(x) \\ = & \quad \textstyle\sum_{x \in A_x} - \log_2(2^{-(x+1)}) \cdot p_x(x) \\ = & \quad \textstyle\sum_{x \in A_x} (x+1) \cdot p_x(x) \qquad = R \end{aligned}$$

c) Golomb codewords:

symbol x	Golomb codeword c_x
0	10
1	11
2	010
3	011
4	0010
5	0011

Table 5: Golomb Coding

d) Decoding:

The first symbol ($x_q = 1$ and $x_r = 4$) is decoded to $x = 16 \cdot 1 + 4 = 20$.
The second symbol ($x_q = 1$ and $x_r = 4$) is decoded to $x = 16 \cdot 1 + 4 = 20$.
Concatenating the decoded symbols results in “2020”.

2.2 Huffman Coding

a) Mean and variance:

Mean value:
$\mathrm{E}\left(X\right) = \mu = \sum_{x \in A_x} x \cdot p_x\left(x\right) = 1.59$

Variance:
$\sigma^2 = \sum_{x \in A_x} p_x\left(x\right)\left(x-\mu\right)^2 = 0.4219$

b) Self information and entropy:

symbol	probability	Self information $h_x(x)$
1	0.5	1
2	0.41	1.2863
3	0.09	3.4739

Table 6: Symbol probabilities

$$H\left(X\right) = -\sum_{x \in A_x} p_x\left(x\right) \cdot \log_2\left(p_x\left(x\right)\right) = \mathrm{E}\left\{h_x\left(x\right)\right\} = \sum_{x \in A_x} p_x\left(x\right) \cdot h_x\left(x\right) \approx 1.34 \left[\frac{\text{bit}}{\text{symbol}}\right]$$

c) Huffman code:

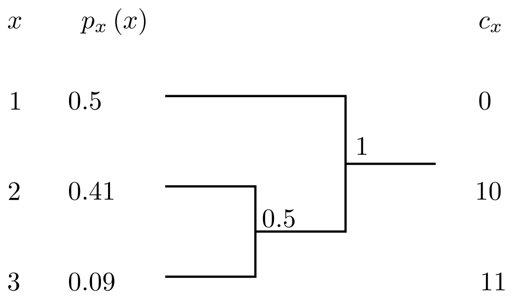


Figure 1: Huffman code

Coding the symbols 1-2-3-1-2-1-2-1: 0 10 11 0 10 0 10 0

d) Lower/upper bound, average code word length and coding redundancy:

$H(X) \leq R \leq H(X) + 1 \rightarrow 1.34 \leq R \leq 2.34$

The average code word length is: $R = \sum_{x \in A_x} p_x\left(x\right) L_x = 1.5 \left[\frac{\text{bit}}{\text{symbol}}\right]$

redundancy: $r = R - H\left(X\right) = 0.16 \left[\frac{\text{bit}}{\text{symbol}}\right]$

e) Vector Huffman code:

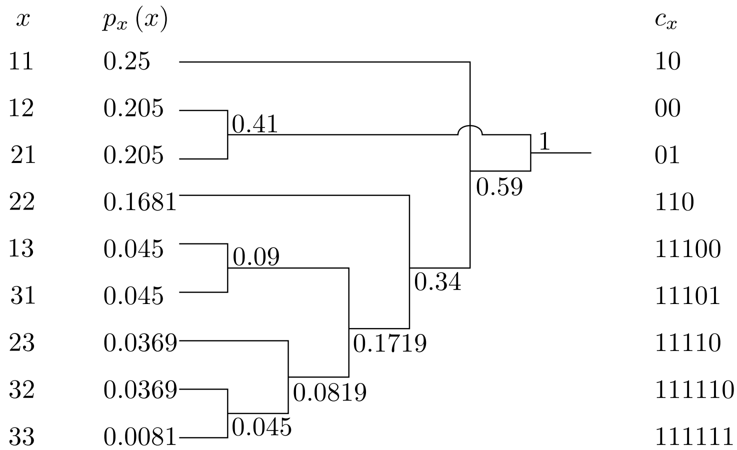


Figure 2: Huffman code

Coding the symbols 1-2-3-1-2-1-2-1: 00 11100 01 01

f) Redundancy per hyper-symbol and the redundancy per symbol:

The average code word length is: $R = \sum_{x \in A_x} p_x(x) L_x = 2.7288 \left[\frac{\text{bit}}{\text{hyper-symbol}}\right]$

redundancy: $r = R - 2H(X) = 0.0488 \left[\frac{\text{bit}}{\text{hyper-symbol}}\right]$

$\Rightarrow \frac{0.0488}{2} \left[\frac{\text{bit}}{\text{symbol}}\right] = 0.0244 \left[\frac{\text{bit}}{\text{symbol}}\right]$

2.3 Arithmetic Coding

a) Expected code word length:

Expected code word length in bits for this example:

$$H(\mathbf{X}) \leq R \leq H(\mathbf{X}) + 2$$

Probability of symbol string: $P("6090") = p_x(6) \cdot p_x(0) \cdot p_x(9) \cdot p_x(0) = 0.15 \cdot 0.5 \cdot 0.17 \cdot 0.5 = 0.006375$

$$\Longrightarrow 7.29 \leq R \leq 9.29$$

So we need 8 or 9 bits.

b) Coding:

Calculation of the cumulative probabilities q_i:

i	symbol x_i	Interval I	p_i	q_i
0	0	[0 ; 0.5)	0.5	0
1	2	[0.5 ; 0.61)	0.11	0.5
2	4	[0.61 ; 0.64)	0.03	0.61
3	6	[0.64 ; 0.79)	0.15	0.64
4	8	[0.79 ; 0.83)	0.04	0.79
5	9	[0.83 ; 1)	0.17	0.83

Table 7: Division of the interval [0..1)

Coding	lower bound	upper bound	width
Initialization	$l_0 = 0$	$u_0 = 1$	$d_0 = 1$
Coding '6' $n = 1, i = 3$	$l_1 = l_0 + d_0 \cdot q_3 = 0 + 1 \cdot 0.64 = 0.64$	$u_1 = l_1 + d_1 = 0.79$	$d_1 = d_0 \cdot p_3 = $ $= 1 \cdot 0.15 = 0.15$
Coding '0' $n = 2, i = 0$	$l_2 = l_1 + d_1 \cdot q_0 = 0.64 + 0.15 \cdot 0 = 0.64$	$u_2 = l_2 + d_2 = 0.715$	$d_2 = d_1 \cdot p_0 =$ $= 0.15 \cdot 0.5 = 0.075$
Coding '9' $n = 3, i = 5$	$l_3 = l_2 + d_2 \cdot q_5 = 0.64 + 0.075 \cdot 0.83 = 0.70225$	$u_3 = l_3 + d_3 = 0.715$	$d_3 = d_2 \cdot p_5$ $= 0.075 \cdot 0.17 = 0.01275 =$
Coding '0' $n = 4, i = 0$	$l_4 = l_3 + d_3 \cdot q_0 = 0.70225 + 0.01275 \cdot 0 = 0.70225$	$u_4 = l_4 + d_4 = 0.708625$	$d_4 = d_3 \cdot p_0 =$ $= 0.01275 \cdot 0.5 = 0.006375$

- Step 1: Submission of the bits for the binary intervals where the calculated interval is definitely enclosed.

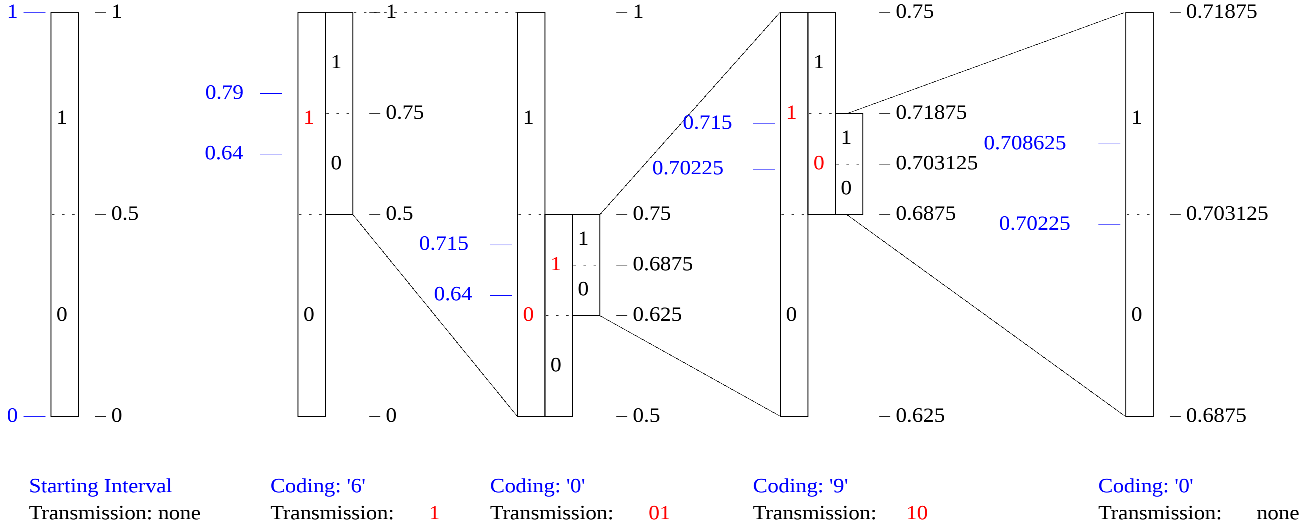


Figure 3: Arithmetic Coding

- Step 2: Narrowing the binary interval until it is non-ambiguous, i.e. unique.

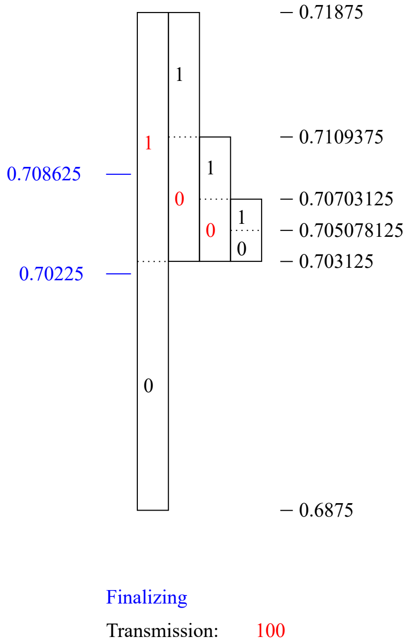


Figure 4: Finalization

Final arithmetic codeword: 10110 100.

c) Decoding 1000011111

Decimal representation for the given sequence:

$$
\begin{aligned}
& 2^{-1}+2^{-6}+2^{-7}+2^{-8}+2^{-9}+2^{-10} \\
= & \frac{1}{2}+\frac{1}{64}+\frac{1}{128}+\frac{1}{256}+\frac{1}{512}+\frac{1}{1024} \\
= & \frac{543}{1024} \approx 0.5302734375
\end{aligned}
$$

Compare with the given probabilities ...
$\in [0.5;\ 0.61) \Rightarrow$ symbol '2', $d = 0.11$
rescale: $\left(\frac{543}{1024} - 0.5\right)/0.11 = \frac{775}{2816} \approx 0.2752$

$\in [0;\ 0.5) \Rightarrow$ symbol '0', $d = 0.5$
rescale: $\left(\frac{775}{2816} - 0\right)/0.5 = \frac{775}{1408} \approx 0.5504$

$\in [0.5;\ 0.61) \Rightarrow$ symbol '2', $d = 0.11$
rescale: $\left(\frac{775}{1408} - 0.5\right)/0.11 = \frac{1775}{3872} \approx 0.4584$

$\in [0;\ 0.5) \Rightarrow$ symbol '0'

stop as only 4 symbols were transmitted
Decoding 1000011111: "2020"

3 Quantization and Entropy Coding

3.1 Quantization

a) Midtread and a midrise quantizer:

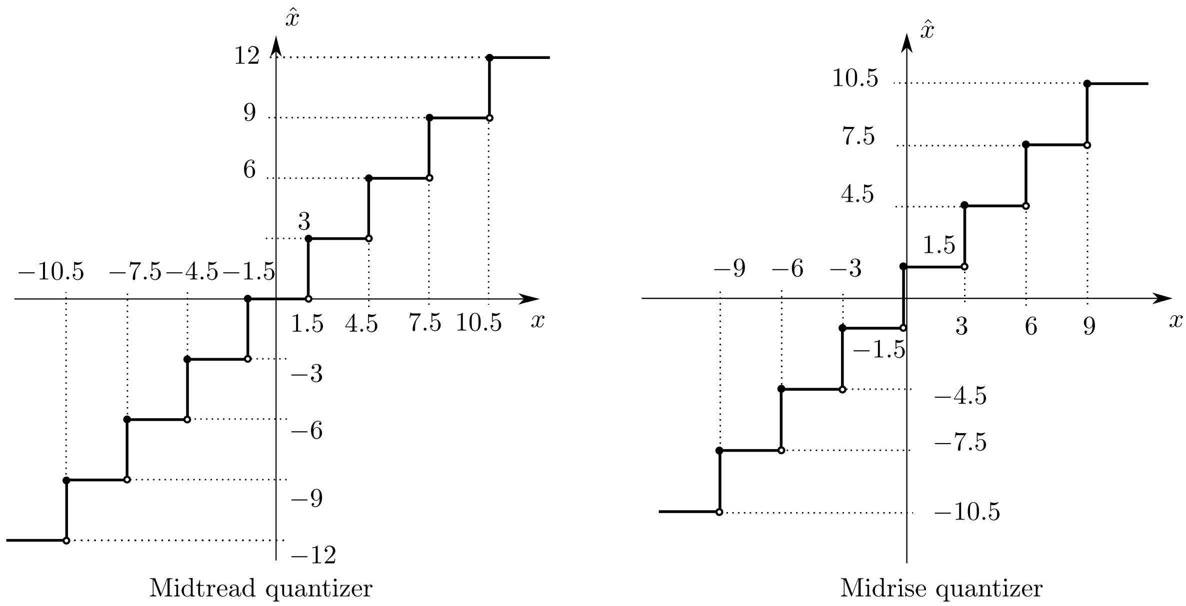


Figure 5: Midtread and midrise quantizer, stepsize 3

$$\boldsymbol{Y}_{\text{midtread,values}} = \begin{bmatrix} -3 & 0 & 0 & 0 & -3 \\ 9 & 0 & 0 & -6 & 0 \\ 6 & 0 & 3 & -9 & 0 \end{bmatrix}$$

$$\boldsymbol{Y}_{\text{midrise,values}} = \begin{bmatrix} -1.5 & 1.5 & -1.5 & 1.5 & -1.5 \\ 10.5 & -1.5 & 1.5 & -7.5 & 1.5 \\ 7.5 & 1.5 & 4.5 & -7.5 & -1.5 \end{bmatrix}$$

b) Better suited quantizer and possible improvement:

Obviously, the midtread quantizer is better suited for the given case because all values around 0 are quantized to zero. An enhancement would be achieved by using a quantizer with deadzone.

c) Two iterations of Lloyd-Max algorithm:

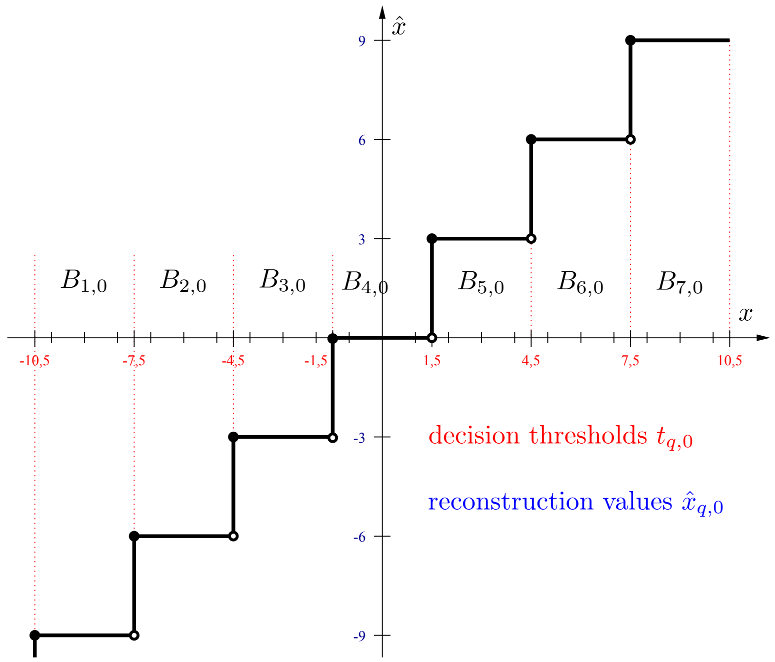


Figure 6: Initialization of the Llyod Max algorithm

$t_{q,i}$	sets $B_{q,i}$	set probabilities $p_x(B_{q,i})$	$\hat{x}_{q,i}$
$t_{1,0} = -10.5$	$B_{1,0} = \{-10, -9, -8\}$	$p_x(B_{1,0}) = \frac{3}{100}$	$\hat{x}_{1,0} = -9$
$t_{2,0} = -7.5$	$B_{2,0} = \{-7, -6, -5\}$	$p_x(B_{2,0}) = \frac{12}{100}$	$\hat{x}_{2,0} = -6$
$t_{3,0} = -4.5$	$B_{3,0} = \{-4, -3, -2\}$	$p_x(B_{3,0}) = \frac{21}{100}$	$\hat{x}_{3,0} = -3$
$t_{4,0} = -1.5$	$B_{4,0} = \{-1, 0, 1\}$	$p_x(B_{4,0}) = \frac{28}{100}$	$\hat{x}_{4,0} = 0$
$t_{5,0} = 1.5$	$B_{5,0} = \{2, 3, 4\}$	$p_x(B_{5,0}) = \frac{21}{100}$	$\hat{x}_{5,0} = 3$
$t_{6,0} = 4.5$	$B_{6,0} = \{5, 6, 7\}$	$p_x(B_{6,0}) = \frac{12}{100}$	$\hat{x}_{6,0} = 6$
$t_{7,0} = 7.5$	$B_{7,0} = \{8, 9, 10\}$	$p_x(B_{7,0}) = \frac{3}{100}$	$\hat{x}_{7,0} = 9$
$t_{8,0} = 10.5$			

Table 8: Lloyx Max, initialization, $i = 0$

Iteration 1:
Step A: Put the new decision thresholds in the middle between the representatives $t_{q,i} = \frac{\hat{x}_{q-1,i-1} + \hat{x}_{q,i-1}}{2}$.
Keep the thresholds at the left border and the right border fixed, so $t_{1,1} = t_{1,0} = -10.5$, and $t_{8,1} = t_{8,0} = 10.5$.
No change, so $t_{q,1} = t_{q,0}$.

Step B: Calculate new representatives from the centroid of the set $\hat{x}_{q,i} = \frac{\int_{t_{q,i}}^{t_{q+1,i}} x \cdot p_x(x)\,\mathrm{d}x}{\int_{t_{q,i}}^{t_{q+1,i}} p_x(x)\,\mathrm{d}x}$:

$$\hat{x}_{1,1} = \frac{-10 \cdot \frac{0}{100} - 9 \cdot \frac{1}{100} - 8 \cdot \frac{2}{100}}{\frac{3}{100}} = -\frac{25}{3} \approx -8.33$$

$$\hat{x}_{2,1} = -\frac{70}{12} \approx -5.83$$

$$\hat{x}_{3,1} = -\frac{61}{21} \approx -2.91$$

$$\hat{x}_{4,1} = 0$$

The positive and negative sides are symmetric, so $\hat{x}_{5,1} = -\hat{x}_{3,1}$, $\hat{x}_{6,1} = -\hat{x}_{2,1}$, and $\hat{x}_{7,1} = -\hat{x}_{1,1}$

Iteration 2:
Step A: Put the decision levels in the middle between the representatives:
Keep the thresholds at the left border and the right border fixed, so $t_{1,2} = t_{1,1} = -10.5$, and $t_{8,2} = t_{8,1} = 10.5$.

$$t_{2,2} = \frac{-\frac{25}{3} + (-\frac{70}{12})}{2} = -\frac{85}{12} \approx -7.08$$
$$t_{3,2} = -\frac{367}{84} \approx -4.37$$
$$t_{4,2} = -\frac{61}{42} \approx -1.45$$

The positive and negative sides are symmetric, so $t_{7,2} = -t_{2,2}$, $t_{6,2} = -t_{3,2}$, and $t_{5,2} = -t_{4,2}$
Step B: Calculate new representatives from the centroid of the area: No change, because no change in set selection.
The Lloyd-Max algorithm terminates here.

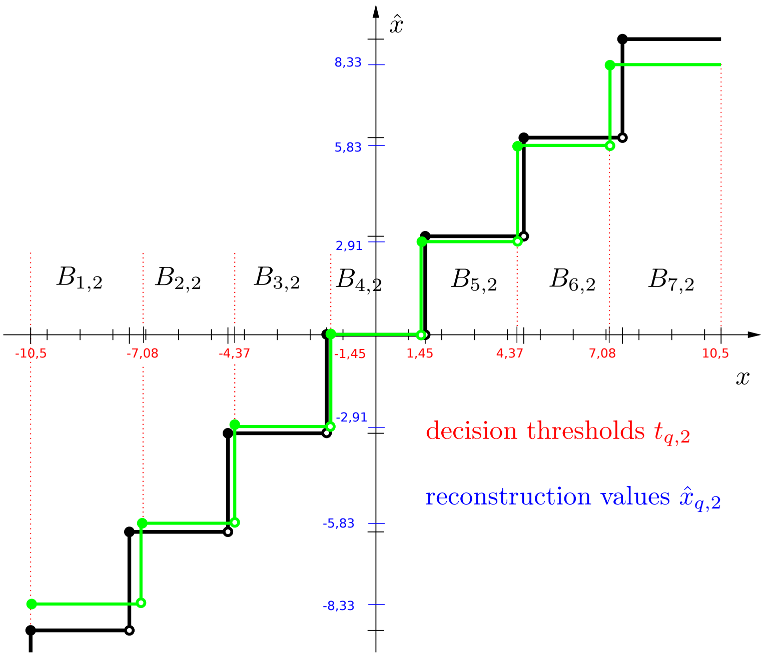


Figure 7: Lloyd Max quantizer after the 2nd iteration (green), uniform midtread quantizer (black)

d) Variance of the quantization error:

mean: $\bar{e} = \sum_{q=1}^{M} p_x(x_q) \cdot e(x_q) = \sum_{q=1}^{M} p_x(x_q) \cdot (x_q - \hat{x}_q)$
variance: $\sigma_e^2 = \sum_{q=1}^{M} p_x(x_q)(e(x_q) - \bar{e})^2$
Uniform midtread quantizer: $\bar{e} = 0$, $\sigma_e^2 = 0.66$
Lloyd Max quantizer: $\bar{e} = 0$, $\sigma_e^2 = 0.62$
The error variance is a little lower.

3.2 Quantization followed by Entropy Coding

a) Line-by-line scan:

$$\boldsymbol{Y}_{\text{midtread,index}} = \begin{bmatrix} -1 & 0 & 0 & 0 & -1 \\ 3 & 0 & 0 & -2 & 0 \\ 2 & 0 & 1 & -3 & 0 \end{bmatrix}$$

$$\boldsymbol{Y}_{\text{midtread,values}} = \begin{bmatrix} \text{-3} & 0 & 0 & 0 & \text{-3} \\ 9 & 0 & 0 & \text{-6} & 0 \\ 6 & 0 & 3 & \text{-9} & 0 \end{bmatrix}$$

Figure 8: Line-by-line scan

line-by-line scan: -3 0 0 0 -3 0 -6 0 0 9 6 0 3 -9 0

Run	0	3	1	2	0	1	0
Value	-3	-3	-6	9	6	3	-9
Idx	-1	-1	-2	3	2	1	-3
Codeword	101	011011	0101001	00000011100	11110	1100	0101011
length	3	6	7	11	5	4	7

Table 9: Run-level coding

Number of Bits: 43

b) Huffman code, required Bits and comparing:

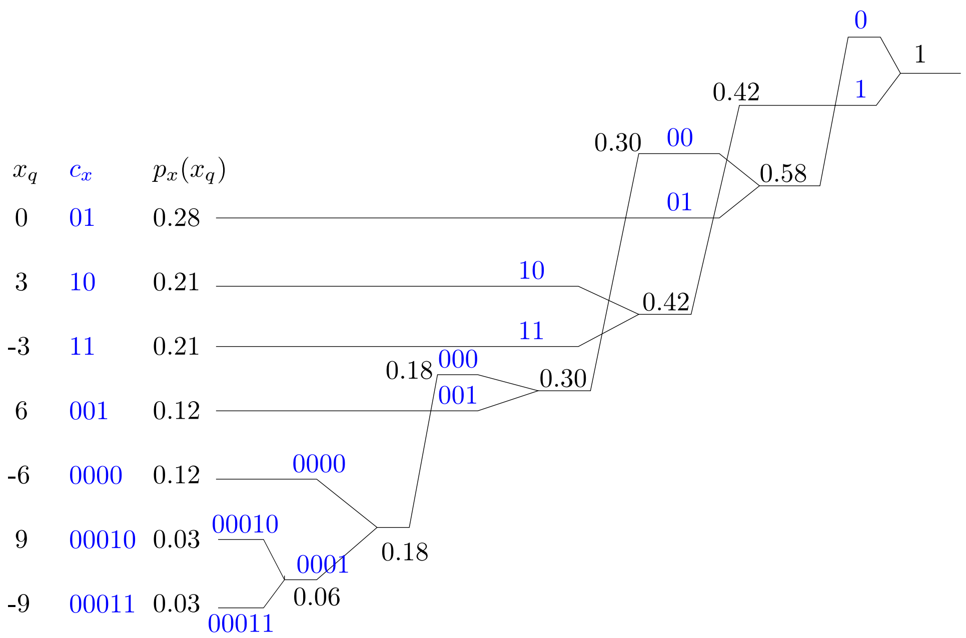


Figure 9: Huffman code

$\Longrightarrow 2 \cdot 11 + 3 \cdot 1 + 4 \cdot 1 + 5 \cdot 2 = 39$ Bits.

The Huffman code for this special signal results in less bits than the run-level-code but the run-level code contains more quantization intervals.

3.3 Vector Quantization

a) Partitions:

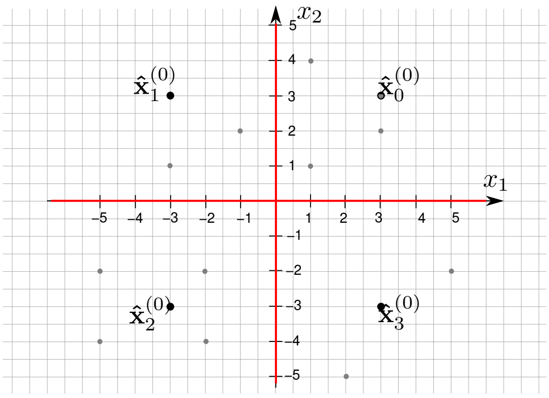


Figure 10: Partitions for the initial reconstruction vectors

b) First iteration of LBG algorithm, B_q and $\hat{\mathbf{x}}_q^{(1)}$:

for $q = 0$:
$B_0 = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_7\} = \{(1,1)^T, (1,4)^T, (3,2)^T, (3,3)^T\}$
$\hat{\mathbf{x}}_0^{(1)} = \frac{B_0}{4} = \frac{1}{4}\begin{pmatrix} 8 \\ 10 \end{pmatrix} = \begin{pmatrix} 2 \\ 2.5 \end{pmatrix}$
for $q = 1$:
$B_1 = \{\mathbf{x}_4, \mathbf{x}_6\} = \{(-1,2)^T, (-3,1)^T\}$
$\hat{\mathbf{x}}_1^{(1)} = \begin{pmatrix} -2 \\ 1.5 \end{pmatrix}$
for $q = 2$:
$B_2 = \{\mathbf{x}_5, \mathbf{x}_{10}, \mathbf{x}_{11}, \mathbf{x}_{12}\} = \{(-2,-2)^T, (-5,-4)^T, (-2,-4)^T, (-5,-2)^T\}$
$\hat{\mathbf{x}}_2^{(1)} = \begin{pmatrix} -3.5 \\ -3 \end{pmatrix}$
for $q = 3$:
$B_3 = \{\mathbf{x}_8, \mathbf{x}_9\} = \{(5,-2)^T, (2,-5)^T\}$
$\hat{\mathbf{x}}_3^{(1)} = \begin{pmatrix} 3.5 \\ -3.5 \end{pmatrix}$

c) Partitions after the first iteration of LBG algorithm:

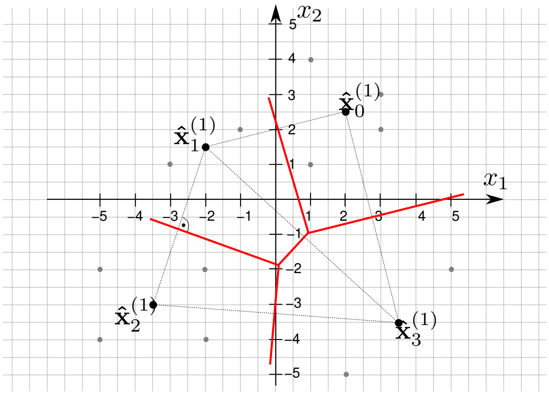


Figure 11: Partitions after the first iteration of the LBG algorithm

4 Linear Prediction and DPCM

a) Block diagram:

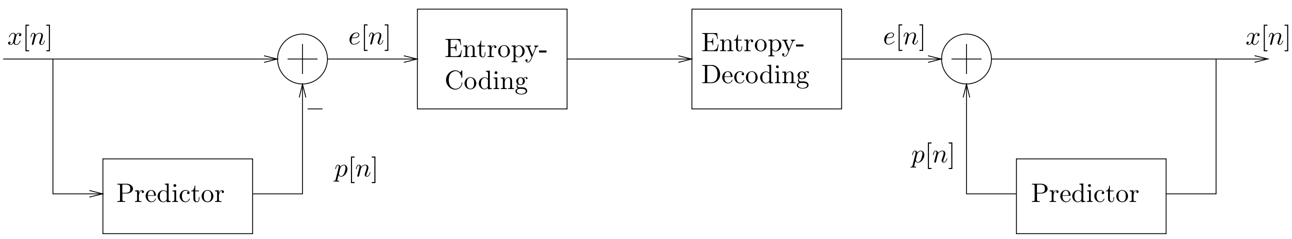


Figure 12: Block diagram for a predictive coder and decoder. (See slice 6-2 in the lecture notes)

b) Block diagram:

$$p[n] = \sum_{i=1}^{3} a_i x[n-i]$$

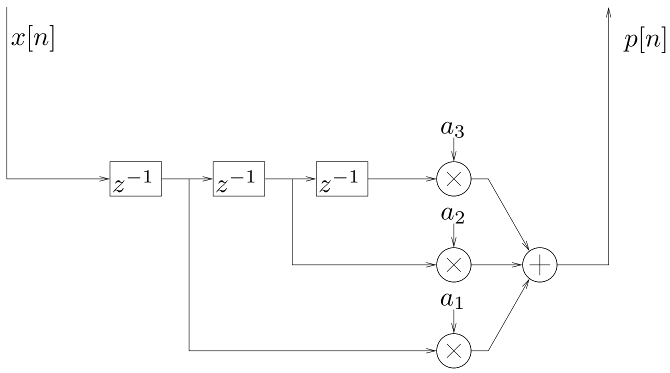


Figure 13: Block diagram of a linear predictor

c) Wiener-Hopf equation:

$\boldsymbol{\varphi}_{xx} = \boldsymbol{\Phi}_{xx}\boldsymbol{a}$, with

$$\boldsymbol{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}, \quad \boldsymbol{\Phi}_{xx} = \begin{bmatrix} \varphi_{xx}[0] & \varphi_{xx}[1] & \varphi_{xx}[2] \\ \varphi_{xx}[1] & \varphi_{xx}[0] & \varphi_{xx}[1] \\ \varphi_{xx}[2] & \varphi_{xx}[1] & \varphi_{xx}[0] \end{bmatrix}, \quad \boldsymbol{\varphi}_{xx} = \begin{bmatrix} \varphi_{xx}[1] \\ \varphi_{xx}[2] \\ \varphi_{xx}[3] \end{bmatrix}$$

The weights can be computed by $\boldsymbol{a} = \boldsymbol{\Phi}_{xx}^{-1}\boldsymbol{\varphi}_{xx}$.

d) First two elements:

$$\begin{aligned}\varphi_{xx}[0] &= \tfrac{1}{15}\textstyle\sum_{n=0}^{14} x[n]\cdot x[n-0] = \tfrac{1}{15} \quad \big(8^2+2^2+(-4)^2+(-8)^2+(-16)^2+(-20)^2+0^2+ \\ &\qquad +0^2+0^2+0^2+0^2+4^2+8^2+12^2+16^2\big) = 85.6\end{aligned}$$

$$\begin{aligned}\varphi_{xx}[1] &= \tfrac{1}{14}\textstyle\sum_{n=1}^{14} x[n]\cdot x[n-1] = \tfrac{1}{14} \quad (2\cdot 8+(-4)\cdot 2+(-8)(-4)+(-16)(-8)+ \\ &\qquad +(-20)(-16)+0\cdot(-20)+0\cdot 0+0\cdot 0+ \\ &\qquad +0\cdot 0+0\cdot 0+4\cdot 0+8\cdot 4+12\cdot 8+16\cdot 12) \approx 57.7143\end{aligned}$$

e) prediction coefficients a_1, a_2 and a_3:

$$\varphi_{xx}[2] = \frac{1}{13}\sum_{n=2}^{14} x[n]\cdot x[n-2] \approx 27.0769$$

$$\varphi_{xx}[3] = \frac{1}{12}\sum_{n=3}^{14} x[n]\cdot x[n-3] = 4$$

$$\boldsymbol{a} = \boldsymbol{\Phi}^{-1}\boldsymbol{\varphi} = \begin{bmatrix} 0.0229 & -0.0193 & 0.0058 \\ -0.0193 & 0.0378 & -0.0193 \\ 0.0058 & -0.0193 & 0.0229 \end{bmatrix} \cdot \begin{bmatrix} 57.7143 \\ 27.0769 \\ 4 \end{bmatrix} = \begin{bmatrix} 0.8223 \\ -0.1676 \\ -0.0962 \end{bmatrix}$$

$a_1 = 0.8223$, $a_2 = -0.1676$, $a_3 = -0.0962$

f) First 5 values of $e[n]$:

$e[0] = x[0] - a_1\cdot x[-1] - a_2\cdot x[-2] - a_3\cdot x[-3] = x[0] = 8$
$e[1] = x[1] - a_1\cdot x[0] - a_2\cdot x[-1] - a_3\cdot x[-2] = x[1] - a_1\cdot x[0] \approx 2 - 8\cdot 0.8223 \approx -4.5782$
$e[2] \approx -4.3039$
$e[3] \approx -3.6058$
$e[4] \approx -9.8996$

g) Decoding:

$x[0] = e[0] + a_1\cdot x[-1] + a_2\cdot x[-2] + a_3\cdot x[-3] = e[0] = 8$
$x[1] = e[1] + a_1\cdot x[0] + a_2\cdot x[-1] + a_3\cdot x[-2] = e[1] + a_1\cdot x[0] = -4.5782 + 0.8223\cdot 8 \approx 2$
$x[2] \approx -4$
$x[3] \approx -8$
$x[4] \approx -16$

h) Explanation:

No, you can't. The previously decoded values are needed to decode the current signal. Otherwise the decoder will drift. In this example the previous values were initialized to 0.

i) Block diagram:

That's the Difference Pulse Code Modulation (DPCM):

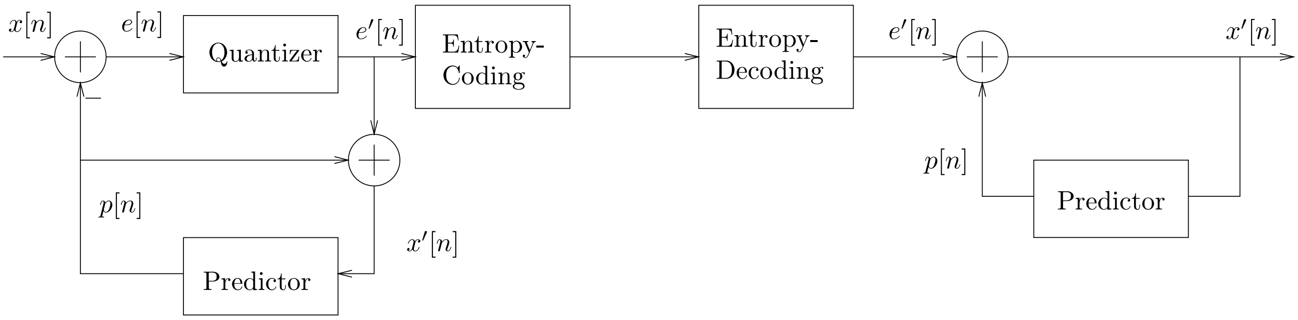


Figure 14: DPCM coder and decoder

j) Mark the hidden decoder:

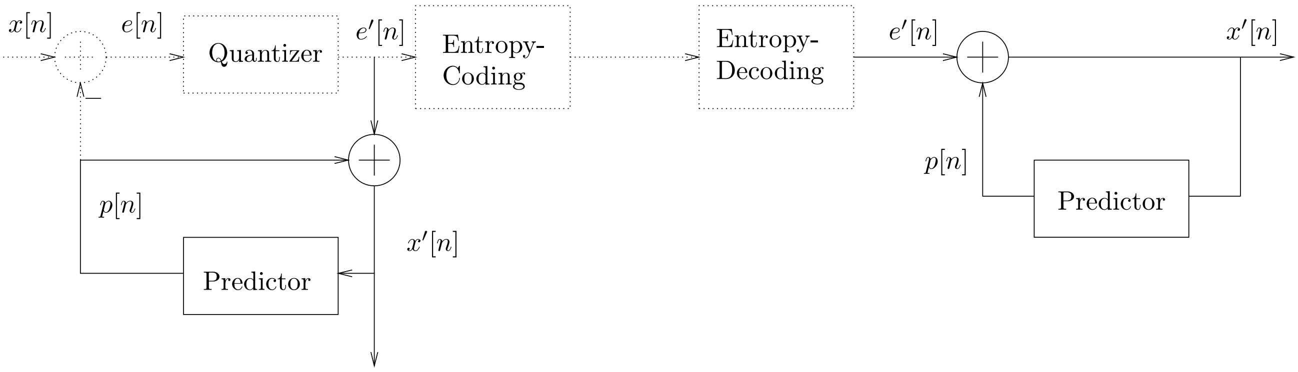


Figure 15: DPCM, decoder marked

k) $e'[n]$ and $x'[n]$:

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$x[n]$	8	2	-4	-8	-16	-20	0	0	0	0	0	4	8	12	16
$p[n]$	0	9	3	-3	-9	-15	-21	-12	-3	0	0	0	3	9	12
$e[n]$	8	-7	-7	-5	-7	-5	21	12	3	0	0	4	5	3	4
$e'[n]$	9	-6	-6	-6	-6	-6	9	9	3	0	0	3	6	3	3
$x'[n]$	9	3	-3	-9	-15	-21	-12	-3	0	0	0	3	9	12	15

l) Artifacts:

From 6-9 we can see a 'slope overload' and at the end from 11 to 14 we can see a faint example of 'edge business'.

5 Transforms

5.1 General questions about transforms

a) Advantages of the DCT when compared to the DFT:

It contains only real values and there are no discontinuities near the borders.

b) Karhunen-Loéve Transform, Advantages and disadvantages:

Compute the eigenvectors $\boldsymbol{v}_k$ and the eigenvalues λ_k of $\boldsymbol{\Psi}_{xx} = \begin{bmatrix} \sigma_x^2 & \psi_{xx}[1] & \cdots & \psi_{xx}[N-1] \\ \psi_{xx}[1] & \sigma_x^2 & & \psi_{xx}[N-2] \\ \vdots & & \ddots & \vdots \\ \psi_{xx}[N-1] & \psi_{xx}[N-2] & \cdots & \sigma_x^2 \end{bmatrix}$

$\rightarrow \boldsymbol{\Psi}_{xx}\boldsymbol{v}_k = \lambda_k \boldsymbol{v}_k$, for $0 \leq k \leq N-1$
$\rightarrow$ KLT transform matrix $\boldsymbol{V} = [\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_{N-1}]$
$\rightarrow$ KLT is computed by $\boldsymbol{y} = \boldsymbol{V}\boldsymbol{x}$

Advantages: Optimal energy compaction, optimal decorrelation.

Disadvantage: Depends on the signal, computationally complex

5.2 Separable Transform

a) transformed signal $\boldsymbol{Y}$

$\boldsymbol{Y} = \boldsymbol{A} \cdot \boldsymbol{X} \cdot \boldsymbol{A}^T$

b) Main advantage:

The complexity is decreased from $O(N^4)$ to $O(N^3)$

c) Orthogonality:

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} = (a_0, a_1, \ldots, a_{N-1})^\intercal$$

The transform is orthogonal if: $a_k a_l^\intercal = \begin{cases} \neq 0 & \text{for } k = l \\ = 0 & \text{for } k \neq l \end{cases}$

$a_0 a_1^\intercal = 1 \cdot 2 + 1 \cdot 1 + 1 \cdot (-1) + 1 \cdot (-2) = 0$
$a_0 a_2^\intercal = 0$
$a_0 a_3^\intercal = 0$
$a_1 a_2^\intercal = 0$
$a_1 a_3^\intercal = 0$
$a_2 a_3^\intercal = 0$

d) Orthonormality:

The transform is orthonormal if: $a_k a_l^\intercal = \begin{cases} 1 & \text{for } k = l \\ 0 & \text{for } k \neq l \end{cases}$

$a_0 a_0^\intercal = 1^2 + 1^2 + 1^2 + 1^2 = 4$
$a_1 a_1^\intercal = 2^2 + 1^2 + (-1)^2 + (-2)^2 = 10$
$a_2 a_2^\intercal = 1^2 + (-1)^2 + (-1)^2 + 1^2 = 4$
$a_3 a_3^\intercal = 1^2 + (-2)^2 + 2^2 + (-1)^2 = 10$

Because the sums of the squared elements are not equal to 1, the transform is not orthonormal but we can change it by division of each row by the root of its sum of the squared elements.

$$A_{\text{orthonormal}} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{2}{\sqrt{10}} & \frac{1}{\sqrt{10}} & -\frac{1}{\sqrt{10}} & -\frac{2}{\sqrt{10}} \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\ \frac{1}{\sqrt{10}} & -\frac{2}{\sqrt{10}} & \frac{2}{\sqrt{10}} & -\frac{1}{\sqrt{10}} \end{bmatrix}$$

e) Missing values:

$$\boldsymbol{Y}_C = \boldsymbol{A} \cdot \boldsymbol{X} = \begin{bmatrix} 132.5 & \boxed{338.5} & 345.5 & 126 \\ -85.6977 & 54.7074 & 84.1166 & -62.2969 \\ 5.5 & \boxed{9.5} & 1.5 & -13 \\ -12.0167 & 10.7517 & 19.9223 & -19.2899 \end{bmatrix}$$

$$\boldsymbol{Y} = \boldsymbol{Y}_C \cdot \boldsymbol{A}^T = \begin{bmatrix} 471.25 & 1.89737 & \boxed{\text{-212.75}} & 6.48267 \\ -4.5853 & -24.1 & -143.409 & 11.2 \\ 1.75 & 14.2302 & -9.25 & 0.790569 \\ -0.316228 & 1.7 & -30.9903 & 8.1 \end{bmatrix}$$

f) Quantizer indices (levels) and the quantized signal:

$$Y_{Q\text{index}}(x, y) = \text{round}\left(\frac{Y(x,y)}{Q(x,y)}\right)$$

$$\boldsymbol{Y}_{Q\text{index}} = \begin{bmatrix} 118 & 0 & -13 & 0 \\ -1 & -2 & -4 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad \boldsymbol{Y}_Q = \begin{bmatrix} 472 & 0 & -208 & 0 \\ -8 & -32 & -128 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

g) Equation for inverse transformation:

$\boldsymbol{X}_Q = \boldsymbol{A}^T \cdot \boldsymbol{Y}_Q \cdot \boldsymbol{A}$

5.3 Discrete Cosine Transform

a) Missing transform coefficients:

DCT:

$$F(u,v) = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} c(u)\cdot c(v)\cdot f(x,y)\cdot\cos\left(\frac{(2x+1)\cdot u\cdot\pi}{2\cdot N}\right)\cdot\cos\left(\frac{(2y+1)\cdot v\cdot\pi}{2\cdot N}\right)$$

with

$$c(i) = \begin{cases}\sqrt{\frac{1}{N}} & i=0\\ \sqrt{\frac{2}{N}} & i>0\end{cases}$$

compute

$$\begin{aligned}
M_{1,T}(0,0) &= c(0)\cdot c(0)\sum_{x=0}^{3}\sum_{y=0}^{3} f(x,y)\cdot\underbrace{\cos\left(\frac{(2x+1)\cdot(u=0)\cdot\pi}{8}\right)}_{=1}\cdot\underbrace{\cos\left(\frac{(2y+1)\cdot(v=0)\cdot\pi}{8}\right)}_{=1}\\
&= \sqrt{\frac{1}{4}}\sqrt{\frac{1}{4}}\sum_{x=0}^{3}\sum_{y=0}^{3}100 = \frac{1}{4}\cdot 16\cdot 100 = 400
\end{aligned}$$

$$\begin{aligned}
M_{1,T}(1,1) &= c(1)\cdot c(1)\sum_{x=0}^{3}\sum_{y=0}^{3} f(x,y)\cdot\cos\left(\frac{(2x+1)\cdot(u=1)\cdot\pi}{8}\right)\cdot\cos\left(\frac{(2y+1)\cdot(v=1)\cdot\pi}{8}\right)\\
&= \sqrt{\frac{2}{4}}\sqrt{\frac{2}{4}}\cdot 100\cdot 0 = 0
\end{aligned}$$

$$\text{transformed matrix 1: } \boldsymbol{M}_{1,T} = \begin{bmatrix} 400 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}$$

$cos\frac{\pi}{8} \cdot cos\frac{\pi}{8}$	$cos\frac{3}{8}\pi \cdot cos\frac{\pi}{8}$	$cos\frac{5}{8}\pi \cdot cos\frac{\pi}{8}$	$cos\frac{7}{8}\pi \cdot cos\frac{\pi}{8}$
$cos\frac{\pi}{8} \cdot cos\frac{3}{8}\pi$	$cos\frac{3}{8}\pi \cdot cos\frac{3}{8}\pi$	$cos\frac{5}{8}\pi \cdot cos\frac{3}{8}\pi$	$cos\frac{7}{8}\pi \cdot cos\frac{3}{8}\pi$
$cos\frac{\pi}{8} \cdot cos\frac{5}{8}\pi$	$cos\frac{3}{8}\pi \cdot cos\frac{5}{8}\pi$	$cos\frac{5}{8}\pi \cdot cos\frac{5}{8}\pi$	$cos\frac{7}{8}\pi \cdot cos\frac{5}{8}\pi$
$cos\frac{\pi}{8} \cdot cos\frac{7}{8}\pi$	$cos\frac{3}{8}\pi \cdot cos\frac{7}{8}\pi$	$cos\frac{5}{8}\pi \cdot cos\frac{7}{8}\pi$	$cos\frac{7}{8}\pi \cdot cos\frac{7}{8}\pi$

Table 10: Table containing the combinations of the cosine terms for $M_{1,T}(1,1)$

$$\begin{aligned}
M_{2,T}(1,0) &= c(0)\cdot c(1)\sum_{x=0}^{3}\sum_{y=0}^{3} f(x,y)\cdot\cos\left(\frac{(2x+1)\cdot 1\cdot\pi}{8}\right)\cdot\underbrace{\cos\left(\frac{(2y+1)\cdot 0\cdot\pi}{8}\right)}_{=1}\\
&= \sqrt{\frac{1}{4}}\sqrt{\frac{2}{4}}\sum_{x=0}^{3}\sum_{y=0}^{3} f(x,y)\cos\left(\frac{(2x+1)\cdot 1\cdot\pi}{8}\right)\\
&= \frac{1}{2\sqrt{2}}\cdot\underbrace{4}_{\text{the same for all y}}\cdot 100\cdot\underbrace{\cos\left(\frac{3}{8}\pi\right)}_{f(x,y)\neq 0\text{ only for }x=1} = 54.1196
\end{aligned}$$

$$
\begin{aligned}
M_{2,T}(3,0) &= c(0)\cdot c(3)\sum_{x=0}^{3}\sum_{y=0}^{3} f(x,y)\cdot\cos\left(\frac{(2x+1)\cdot 3\cdot\pi}{8}\right)\cdot\underbrace{\cos\left(\frac{(2y+1)\cdot 0\cdot\pi}{8}\right)}_{=1} \\
&= \sqrt{\frac{1}{4}}\sqrt{\frac{2}{4}}\sum_{x=0}^{3}\sum_{y=0}^{3} f(x,y)\cos\left(\frac{(2x+1)\cdot 3\cdot\pi}{8}\right) \\
&= \frac{1}{2\sqrt{2}}\cdot\underbrace{4}_{\text{the same for all y}}\cdot 100\cdot\underbrace{\cos\left(\frac{9}{8}\pi\right)}_{f(x,y)\neq 0 \text{ only for } x=1} = -130.6563
\end{aligned}
$$

$$
\text{transformed matrix 2: } \boldsymbol{M}_{2,T} = \begin{bmatrix} 100 & 54.1196 & -100 & -130.6563 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}
$$

b) Quantizer indices (levels):

Quantization:
Element-wise division!

$$
\tilde{F}(x,y) = \text{round}\left(\frac{F(x,y)}{Q(x,y)}\right)
$$

$$
\text{transformed und quantized matrix 1: } \boldsymbol{M}_{1,T,Q\text{index}} = \begin{bmatrix} 44 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}
$$

$$
\text{transformed und quantized matrix 2: } \boldsymbol{M}_{2,T,Q\text{index}} = \begin{bmatrix} 11 & 4 & -5 & -4 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}
$$

c) Quantized matrices:

Element-wise multiplication

$$
F(x,y) = \tilde{F}(x,y)\cdot Q(x,y)
$$

$$
\text{transformed und quantized matrix 1: } \boldsymbol{M}_{1,T,Q} = \begin{bmatrix} 396 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}
$$

$$
\text{transformed und quantized matrix 2: } \boldsymbol{M}_{2,T,Q} = \begin{bmatrix} 99 & 60 & -100 & -120 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}
$$

d) Missing values:

inverse 2-D DCT:

$$f(x,y) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} c(u)\cdot c(v)\cdot F(u,v)\cdot \cos\left(\frac{(2x+1)\cdot u\cdot\pi}{2\cdot N}\right)\cdot\cos\left(\frac{(2y+1)\cdot v\cdot\pi}{2\cdot N}\right)$$

$$\begin{aligned} M_{1,T,Q,T^{-1}}(x,y) &= \sum_{u=0}^{3}\sum_{v=0}^{3} c(u)\cdot c(v)\cdot \underbrace{M_{1,T,Q}(u,v)}_{=396 \text{ for } (u=0)\wedge(v=0)}\cdot \cos\left(\frac{(2x+1)\cdot u\cdot\pi}{2\cdot N}\right)\cdot\cos\left(\frac{(2y+1)\cdot v\cdot\pi}{2\cdot N}\right) \\ &= c(0)\cdot c(0)\cdot M_{1,T,Q}(0,0)\cdot 1\cdot 1 = \frac{1}{4}\cdot 396 = 99 \end{aligned}$$

independent of x and y!

$$\text{inverse transformed and quantized matrix 1: } \boldsymbol{M}_{1,T,Q,T^{-1}} = \begin{bmatrix} 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 \end{bmatrix}$$

$$\begin{aligned} M_{2,T,Q,T^{-1}}(x,y) &= \sum_{u=0}^{3}\sum_{v=0}^{3} c(u)\cdot c(v)\cdot \underbrace{M_{2,T,Q}(u,v)}_{=0 \text{ for } (v\neq 0)}\cdot \cos\left(\frac{(2x+1)\cdot u\cdot\pi}{2\cdot N}\right)\cdot\cos\left(\frac{(2y+1)\cdot v\cdot\pi}{2\cdot N}\right) \\ &= \sum_{u=0}^{3} c(u)\cdot c(0)\cdot M_{2,T,Q}(u,0)\cdot\cos\left(\frac{(2x+1)\cdot u\cdot\pi}{8}\right)\cdot 1 \end{aligned}$$

independent of y!
Example for first column: $x=0$

$$\begin{aligned} M_{2,T,Q,T^{-1}}(0,y) &= \sum_{u=0}^{3} c(u)\cdot c(0)\cdot M_{2,T,Q}(u,0)\cdot\cos\left(\frac{(2\cdot 0+1)\cdot u\cdot\pi}{8}\right) \\ &= \sqrt{\frac{1}{4}}\left(\underbrace{c(0)\cdot 99\cdot\cos(0)}_{(u=0)} + \underbrace{c(1)\cdot 60\cdot\cos\left(\frac{\pi}{8}\right)}_{(u=1)} - \underbrace{c(2)\cdot 100\cdot\cos\left(\frac{2\pi}{8}\right)}_{(u=2)} - \underbrace{c(3)\cdot 120\cdot\cos\left(\frac{3\pi}{8}\right)}_{(u=3)}\right) \\ &\approx 3.1 \end{aligned}$$

$$M_{2,T,Q,T^{-1}}(1,y) \approx 97.1$$

$$M_{2,T,Q,T^{-1}}(2,y) \approx 2.43$$

$$M_{2,T,Q,T^{-1}}(3,y) \approx -3.6$$

$$\text{inverse transformed and quantized matrix 2: } \boldsymbol{M}_{2,T,Q,T^{-1}} = \begin{bmatrix} 3.1 & 97.1 & 2.43 & -3.6 \\ 3.1 & 97.1 & 2.43 & -3.6 \\ 3.1 & 97.1 & 2.43 & -3.6 \\ 3.1 & 97.1 & 2.43 & -3.6 \end{bmatrix}$$

e) Error:

- Error Estimation for Matrix $\boldsymbol{M}_1$

$$\boldsymbol{E}_1 = \boldsymbol{M}_1 - \boldsymbol{M}_{1,T,Q,T^{-1}} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\text{SAD}\left(\boldsymbol{M}_1,\, \boldsymbol{M}_{1,T,Q,T^{-1}}\right) = \sum_{x,y} \left| M_1\left(x,y\right) - M_{1,T,Q,T^{-1}}\left(x,y\right)\right| = 16$$

$$\text{SSD}\left(\boldsymbol{M}_1,\, \boldsymbol{M}_{1,T,Q,T^{-1}}\right) = \sum_{x,y} \left(M_1\left(x,y\right) - M_{1,T,Q,T^{-1}}\left(x,y\right)\right)^2 = 16$$

- Error Estimation for Matrix $\boldsymbol{M}_2$

$$\boldsymbol{E}_2 = \boldsymbol{M}_2 - \boldsymbol{M}_{2,Q} = \begin{bmatrix} -3.12 & 2.9 & -2.43 & 3.6 \\ -3.12 & 2.9 & -2.43 & 3.6 \\ -3.12 & 2.9 & -2.43 & 3.6 \\ -3.12 & 2.9 & -2.43 & 3.6 \end{bmatrix}$$

$$\text{SAD}\left(\boldsymbol{M}_2,\, \boldsymbol{M}_{2,T,Q,T^{-1}}\right) = \sum_{x,y} \left| M_2\left(x,y\right) - M_{2,T,Q,T^{-1}}\left(x,y\right)\right| = 12.5 \cdot 4 = 48.2$$

$$\text{SSD}\left(\boldsymbol{M}_2,\, \boldsymbol{M}_{2,T,Q,T^{-1}}\right) = \sum_{x,y} \left(M_2\left(x,y\right) - M_{2,T,Q,T^{-1}}\left(x,y\right)\right)^2 = 4 \cdot (9.7344 + 8.41 + 5.9049 + 12.96) = 148.0372$$

6 Subband-Coding

6.1 General Properties

a) Subband coder and decoder, block diagram

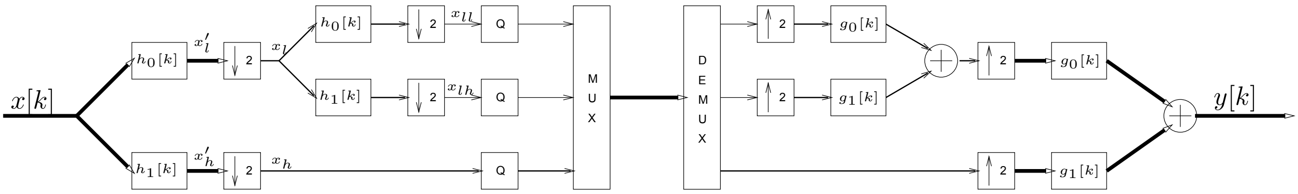


Figure 16: Block diagram for a subband coder and decoder

6.2 Subband decomposition

a) First four signal values:

$$h_0[k] = \frac{1}{8}(-\delta[k+1] + 2\delta[k] + 6\delta[k-1] + 2\delta[k-2] - \delta[k-3])$$
$$h_0[k] = \frac{1}{8}\begin{bmatrix} -1 & 2 & 6 & 2 & -1 \end{bmatrix} \quad \text{with} \quad k = -1, 0, \ldots, 3$$
$$h_1[k] = \frac{1}{2}(-\delta[k+1] + 2\delta[k] - \delta[k-1])$$
$$h_1[k] = \frac{1}{2}\begin{bmatrix} -1 & 2 & -1 \end{bmatrix} \quad \text{with} \quad k = -1, 0, 1$$

- samples of the lowpass band x'_l

$$x'_l[-1] = -20 \cdot (-\tfrac{1}{8}) = 2.5$$
$$x'_l[0] = -20 \cdot \tfrac{2}{8} + (-50) \cdot (-\tfrac{1}{8}) = 1.25$$
$$x'_l[1] = -20 \cdot \tfrac{6}{8} + (-50) \cdot \tfrac{2}{8} + (-30) \cdot (-\tfrac{1}{8}) = -23.75$$
$$x'_l[2] = -20 \cdot \tfrac{2}{8} + (-50) \cdot \tfrac{6}{8} + (-30) \cdot \tfrac{2}{8} + (50) \cdot (-\tfrac{1}{8}) = -56.25$$

- samples of the highpass band x'_h

$$x'_h[-1] = -20 \cdot (-\tfrac{1}{2}) = 10$$
$$x'_h[0] = -20 + (-50) \cdot \left(-\tfrac{1}{2}\right) = 5$$
$$x'_h[1] = -20 \cdot \left(-\tfrac{1}{2}\right) + (-50) + (-30) \cdot \left(-\tfrac{1}{2}\right) = -25$$
$$x'_h[2] = -50 \cdot \left(-\tfrac{1}{2}\right) + (-30) + (50) \cdot \left(-\tfrac{1}{2}\right) = -30$$

b) Coefficients:

$$G_0(z) = z^k H_1(-z)$$
$$G_1(z) = -z^k H_0(-z)$$
for $k = 1$:
$$G_1(z) = -z\frac{1}{8}\left(z^1 + 2 - 6z^{-1} + 2z^{-2} + z^{-3}\right) = \frac{1}{8}\left(-z^2 - 2z + 6 - 2z^{-1} - z^{-2}\right)$$
$$g_1[k] = \frac{1}{8}(-\delta[k+2] - 2\delta[k+1] + 6\delta[k] - 2\delta[k-1] - \delta[k-2])$$
$$g_1[k] = \frac{1}{8}\begin{bmatrix} -1 & -2 & 6 & -2 & -1 \end{bmatrix} \quad \text{with} \quad k = -2, 1, \ldots, 2$$
$$G_0(z) = z\frac{1}{2}\left(z + 2 + z^{-1}\right) = \frac{1}{2}\left(z^2 + 2z\right) + 1$$
$$g_0[k] = \frac{1}{2}(\delta[k+2] + 2\delta[k+1] + \delta[k])$$
$$g_0[k] = \frac{1}{2}\begin{bmatrix} 1 & 2 & 1 \end{bmatrix} \quad \text{with} \quad k = -2, -1, 0$$

c) Standard, the filters are used in:

JPEG 2000 (lossless)

7 Motion Estimation and Color Spaces

7.1 Motion Estimation

a) SAD:

$$\mathrm{SAD}(0,0) = |265-140| + |280-131| + |295-200| + |301-203| = 467$$
$$\mathrm{SAD}(-1,0) = |265-150| + |280-140| + |295-182| + |301-200| = 469$$

So the position (0,0) is better suited, because it has a smaller difference.

b) Explanation:

We end up in 47 and the corresponding motion vector of (4,3). But there are better motion vectors, e.g. 26 at position (-1,2). So the result is not optimal.

7.2 Color Spaces

a) RGB signal:

Yellow: [r g b]=[255 255 0]

$$\begin{pmatrix} y \\ c_b \\ c_r \end{pmatrix} = \begin{pmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{pmatrix} \begin{pmatrix} r \\ g \\ b \end{pmatrix} + \begin{pmatrix} 16 \\ 128 \\ 128 \end{pmatrix} = \begin{pmatrix} 210.0550 \\ 16.0550 \\ 146.1050 \end{pmatrix}$$

rounded to integer values $\begin{pmatrix} y \\ c_b \\ c_r \end{pmatrix} = \begin{pmatrix} 210 \\ 16 \\ 146 \end{pmatrix}$

at the decoder: $\begin{pmatrix} y \\ c_b \\ c_r \end{pmatrix} = \begin{pmatrix} 210 \\ 146 \\ 16 \end{pmatrix}$

$$\begin{pmatrix} r \\ g \\ b \end{pmatrix} = \begin{pmatrix} 1.164 & 0.000 & 1.596 \\ 1.164 & -0.392 & -0.813 \\ 1.164 & 2.017 & 0.000 \end{pmatrix} \begin{pmatrix} 210-16 \\ 146-128 \\ 16-128 \end{pmatrix} = \begin{pmatrix} 47.0640 \\ 309.8160 \\ 262.1220 \end{pmatrix}$$

limited to the co-domain $\begin{pmatrix} r \\ g \\ b \end{pmatrix} = \begin{pmatrix} 47 \\ 255 \\ 255 \end{pmatrix}$

b) Color:

Color vector:

$$\begin{pmatrix} r \\ g \\ b \end{pmatrix} = \begin{pmatrix} 47 \\ 255 \\ 255 \end{pmatrix}$$

Instead of a yellow frame, a cyan frame is decoded.