Introduction to Video Compression H.261

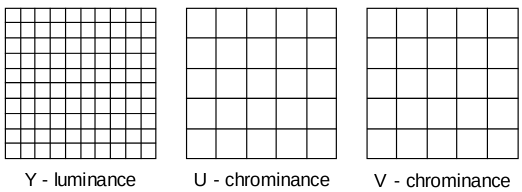
Dirk Farin,

Contact address:
Dirk Farin
University of Mannheim
Dept. Computer Science IV
L 15,16, 68131 Mannheim, Germany
farin@uni-mannheim.de

YUV-Colorspace

- Computer hardware usually uses RGB colorspace for display.
- Video hardware uses YUV colorspace for transmission.
  - Historical reasons: backward compatible to black/white TV.
  - Imitation of human visual system (HVS).
    - Allows better compression.

- Luminance channel Y, two chrominance channels U,V.
- HVS has more receptors for Y than for U,V.
  - Spatial resolution of U,V can be reduced (e.g., 4:2:0)



YUV to RGB conversion

- Since U,V can be negative, an offset of 128 is added.

calculation of greyscale value according to preceived intensities

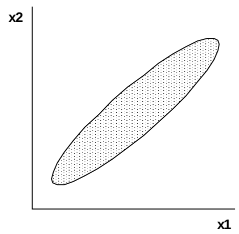
$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.26 & 0.50 & 0.1 \\ -0.15 & -0.29 & 0.44 \\ 0.44 & -0.37 & -0.07 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} 16 \\ 128 \\ 128 \end{pmatrix}$$

- Grey: (Y U V) = (x 128 128)
- Black: (Y U V) = (16 128 128)

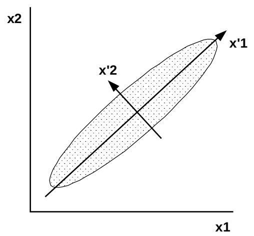
$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1.37 \\ 1 & -0.34 & -0.70 \\ 1 & 1.73 & 0 \end{pmatrix} \begin{pmatrix} Y-16 \\ U-128 \\ V-128 \end{pmatrix}$$

Transform Coding 1

- Exploit correlation between data.



- Independent coding
  - two variables with high entropy



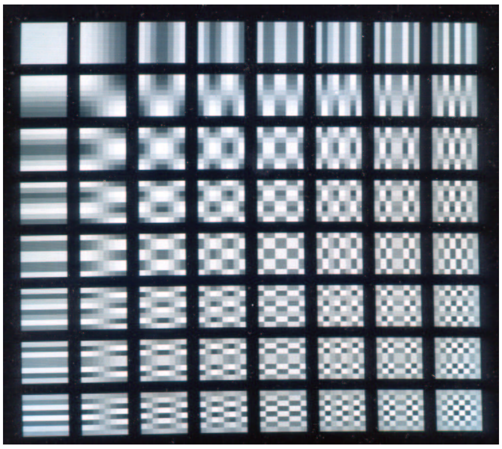
- Decorrelated data
  - only one variable has high entropy

$$\begin{pmatrix} x'_1 \\ x'_2 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

Transform coding / Cosine transform (3)

2-D DCT transform

- cosine wave basis vectors can be used to form 2-D basis images
- 8x8 transform has 64 basis images of 8x8 samples



Dirk Farin, University Mannheim

Transform coding / Cosine transform (1)

2-D Inverse Discrete Cosine Transform (iDCT)

- *Orthogonal* transform

$$f(x,y)=\sum_{u=0}^{7}\sum_{v=0}^{7}\frac{C(u)}{2}\frac{C(v)}{2}\cdot F(u,v)\cdot\cos\left(\frac{(2x+1)u\pi}{16}\right)\cos\left(\frac{(2y+1)v\pi}{16}\right)$$

- *Separable* as two times a 1-D transform, thus

$$f(x,y)=\sum_{v=0}^{7}\frac{C(v)}{2}\left[\sum_{u=0}^{7}\frac{C(u)}{2}\cdot F(u,v)\cdot\cos\left(\frac{(2x+1)u\pi}{16}\right)\right]\cos\left(\frac{(2y+1)v\pi}{16}\right)$$

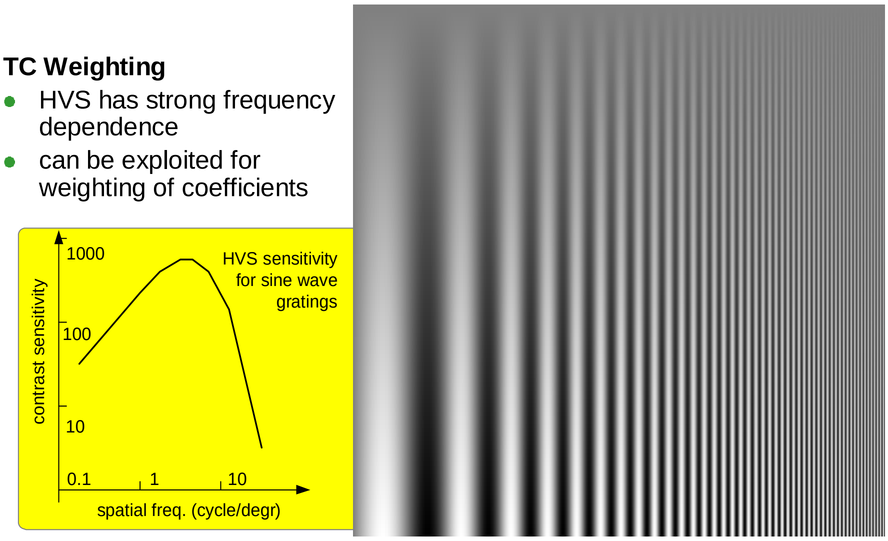
- Implementation as: transform rows => transpose => transform rows

Dirk Farin, University Mannheim

Transform coding / Quantization weighting

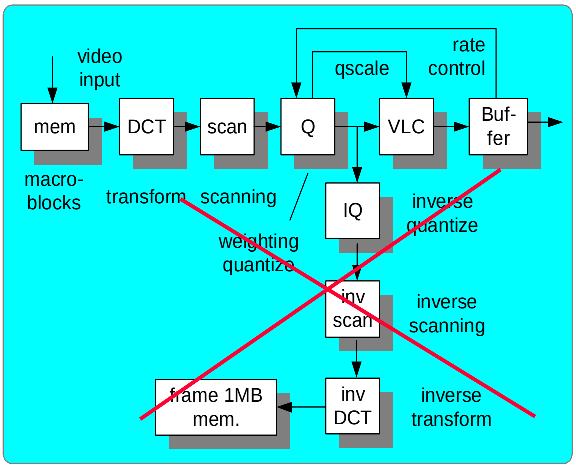
TC Weighting

- HVS has strong frequency dependence
- can be exploited for weighting of coefficients



Intraframe coder

- **intraframe coder/decoder block diagram**
  - local encoding
  - reconstruction for motion compensation



Quantization (1)

MPEG Quantization for inter- / intraframe data

- **DC coefficient**
  - Human eye very sensitive for DC errors**,** thus **fixed** quantizer
  - DC = QDC*8
- **AC coefficients**
  - Weighting W(u,v) according to perception

- H.261:
  - flat AC quantization matrix

MPEG

intra block weighting

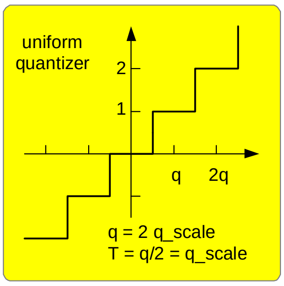
08	16	19	22	26	27	29	34
16	16	22	24	27	29	34	37
19	22	26	27	29	34	34	38
22	22	26	27	29	34	37	40
22	26	27	29	32	35	40	48
26	27	29	32	35	40	48	58
26	27	29	34	38	46	56	69
27	29	35	38	46	56	69	83

inter block weighting

16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16

MPEG Video / Quantization (2)

- **AC coefficients (cont.)**
  - MPEG-1 encoder formula
  - QF(u,v) = 16 F(u,v) / (2 q_scale W(u,v))
  - MPEG-1: decoder formula
  - F(u,v) = 2 (QF(u,v) + k) q_scale W(u,v) / 16
  - k = 0 for intrablocks, and k = sign(QF(u,v)) for non-intra blocks
  - mismatch control (value closest to zero): if F(u,v) even, then F(u,v) = F(u,v) - sign(F(u,v))



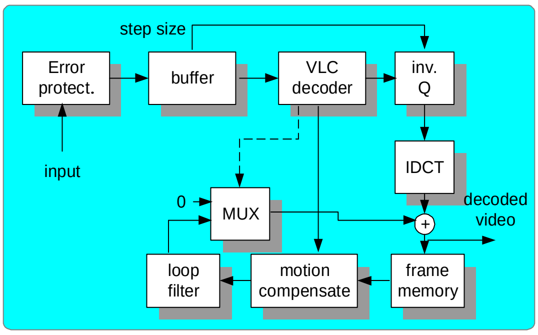
Dirk Farin, University Mannheim

MPEG Video / Quantization (3)

- **MPEG-2 has more precise quantization**
- **DC coefficients**
  - up to 11 bits precision
- **AC coefficients**
  - MPEG-2: decoder formula
  - F(u,v) = 2 (QF(u,v) + k) q_scale W(w,u,v) / 32
  - q_scale is mapped onto larger range than 0...31
  - w is defined by intra / non-intra and colour sampling
  - k = 0 for intrablocks, and k = sign(QF(u,v)) for non-intra blocks
  - special additional mismatch control: F(7,7) = F(7,7) if SUM ac(F(u,v)) is odd, and F(7,7) = F(7,7) +/- 1 if F(7,7) is even/odd and SUM is even.

PA case study 1 / H.261 Video decoder

- Implementation issues
  - specification of DCT computation accuracy
  - error recovery: at least 1intra MB every 132 inter MBs



- only P pictures basically, different E-E delay than in MPEG
- matching of ME search range to temporal frame rate

H.261 Bit-stream syntax

- Only two image sizes:
  - CIF (352x288),
  - QCIF (176x144) quarter-CIF
- Central coding unit: macroblock
  - 16x16 pixels luminance, two times 8x8 pixels chrominance
- Image is divided into groups of macroblocks (GOB).
  - Each GOB has 11x3 MBs.
  - For CIF: 2x6 GOBs, QCIF: 1x3 GOBs
  - Purpose: resynchronization after transmission error.

1	2
3	4
5	6
7	8
9	10
11	12

1
3
5

H.261 Picture Header

- PSC 20 bits Picture Start Code = 0000 0000 0000 0001 0000
- TR 5 bits Temporal Reference
  - continuous frame counter (incremented for next frame), used to code temporal distance between pictures
- PTYPE 6 bits Type information
  - Bit 4: 0 QCIF, 1: CIF
- Extra data
  - while next bit == 1
    - 8 user defined bits follow

H.261 GOB Bit-stream Syntax

- GBSC 16 bits Group of blocks start code
  - = 0000 0000 0000 0001
- GN 4 bits Group number
  - The number of this GOB.
  - Defines spatial position.
  - Note that GN==0 is used for picture header.
- GQUANT 5 bits Quantizer step-size
  - Initial quantizer setting.
- Extra information
  - same as in picture header.

H.261 Macroblock Layer

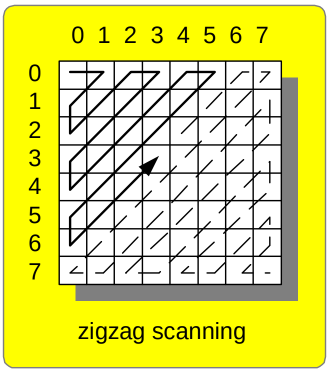
- MBA vlc-1 Increment to get to next MB position
  - not every macroblock has to be coded, MBA>1
- MTYPE vlc-2 Coding type of MB
  - **intra / inter**
  - **mquant ?**
  - motion-vector ?
  - coded-block-pattern ?
  - coefficients ?
  - loop-filter ?
- MQUANT 5 bits New quantizer setting
- MVD vlc Motion vector
- CBP vlc Coded block pattern

H.261 Block-Layer Syntax

- If Intra-block, then DC-coefficient is coded as fixed-length, 8 bits.
- All other coefficients are coded as combined Run/Value pairs.
  - RUN: number of zeros until next non-zero coefficient.
  - LEVEL: value of next coefficient.
  - Special value EOB: End of Block, no more coefficients follow.
- Run/Value pairs are coded with a combined Huffman code.
- Not all combinations are in table.
  - For other combinations, escape-code is used and run/value is coded with fixed length codes.

MPEG Video / Scanning (1)

- **Scanning of transform coefficients**
  - preprocessing step for variable-length coding
  - scanning functions reorders coefficients to cluster zeros for runlength coding
  - start with „low-frequency“ coefficients
  - fundamental scanning pattern is diagonal



MPEG Video / Var.-Length Coding (1)

- **Variable-length coding of AC coefficients: algorithm of (runlength, amplitude) coding**
  - STEP 1: (load coefficient), test of coefficient is zero
  - STEP 2: (update runlength), if zero coefficient, increment zero counter, go to STEP 4
  - STEP 3: (jointly code), if non-zero coefficient, then
  - 3a. jointly code [runlength, amplitude] in one codeword
  - 3b. reset runlength counter
  - STEP 4: (do next coefficient), go to STEP 1. If last coefficient, then go to STEP 5.
  - STEP 5: (EOB) Terminate block with EOB-word, ignore runlength value. Codetable is modified Huffman code.

MPEG Video / Var.-Length Coding (2)

- **2-D VLC table of codewords**
  - unlikely symbols are coded by [escape code]+[fixed suffix]
  - also VLC coding of macroblock address, motion vectors,...

zero run ↓ / amplitude →	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	11	2	3	4	4	5	5	5	6	6	6	7	7	7	8	8	8	8	9	9	14
1	12	4	6	7	7	8	9	9	10	10	10	11	11	12	12	12	12	12	12		
2	12	5	7	9	10	10	11	12	12	13											
3	12	6	8	10	12	12	13														
4	12	6	9	11	12																
5	12	7	10	11	12																
6	13	7	10	12																	
7	13	8	12	12																	
8	13	8	12																		
9	13	9	12																		
10	13	9	13																		

EOB = 4

Example of wordlength table

MPEG Video / Var.-Length Coding (3)

- **2-Dim. VLC table of code-words**

code	runlength	amplitude
10	EOB	
1s (note2)	0	1
11s (note3)	0	1
011s	1	1
0100s	0	2
0101s	2	1
0010 1s	0	3
0011 1s	3	1
0011 0s	4	1
0001 10s	1	2
0001 11s	5	1
0001 01s	6	1
0001 00s	7	1
0000 110s	0	4
0000 100s	2	2
0000 111s	8	1
0000 101s	9	1
0000 01	escape	-

code	runlength	amplitude
0010 0110s	0	5
0010 0001s	0	6
0010 0101s	1	3
0010 0100s	3	2
0010 0111s	10	1
0010 0011s	11	1
0010 0010s	12	1
0010 0000s	13	1
0000 001010s	0	7
0000 001100s	1	4
0000 001011s	2	3
0000 001111s	4	2
0000 001001s	5	2
0000 001110s	14	1
0000 001101s	15	1
0000 001000s	16	1

Note1: s=sign bit, 0=pos/1=neg.
Note 2: code for dct_coeff_first
Note 3: code for dct_coeff_next

Example

- Akiyo 128k:

```
hexadecimal:
00 01 00 9F 00.80 40 00 00 22.E4 4B A2 E8 C0
binary:
00000000 00000001 00000000 10011111 00000000
10000000 01000000 00000000 00000000 00100010
11100100 01001011 10100010 11101000 11000000

00000000 00000001 0000  PSC
0000 1                  Temporal reference = 1
001111                  Type -> CIF
1 00000000              Extra Information
1 00000000
1 00000000
0

0000 0000 0000 0001 0001 GOB-start code  (GOB 1)
011100100 01001011 10100010 11101000 11000000
```

Example

```
00000000 00000001 00000000 10011111 00000000
10000000 01000000 00000000 00000000 00100010
11100100 01001011 10100010 11101000 11000000
01101100 01100000 00110110 01111110 10100000

0000 0000 0000 0001 0001 GOB-start code  (GOB 1)
01110     Gquant = 14
0         no extra information

1         MB addess increment = 1  (vlc)
0001      MB-Type = Intra, no MQUANT

00101110  DC = 46 -> 368 (dequant)
10        EOB
00101110  DC = 46 -> 368
10        EOB
00110000  DC = 48 -> 384
```


Example

```
00000000 00000001 00000000 10011111 00000000
10000000 01000000 00000000 00000000 00100010
11100100 01001011 10100010 11101000 11000000
01101100 01100000 00110110 01111110 10100000

00110000  DC = 48 -> 384
0001101   1 / -2
10        EOB
00110000  DC = 48 -> 384
0001101   1 / -2
10        EOB
01111110  DC = -> 1008
10        EOB

100000   ...
```