Multimedia Video Coding & Architectures (5LSE0), Module 08

Intraframe DCT Coding, Standards JPEG and DV

Peter H.N. de With
(p.h.n.de.with@tue.nl)

slides version 1.0

5LSE0 - Mod 08
Part 1
JPEG Still image compression standard

Introduction to JPEG

- **International standardization in 1988 - 1992 led to ISO and ITU-Telecommunications standard for still picture coding**
  - Referred to as *ISO/IEC IS 10918*
- **JPEG: Joint Photographic Experts Group**
- **Rationales**
  - *Comparable* requirements for *many* applications
  - Possible exchange of compressed data
  - Hardware/software reduction (>10 years ago!)
  - DCT had become very popular
- **JPEG is a strongly prescriptive standard**

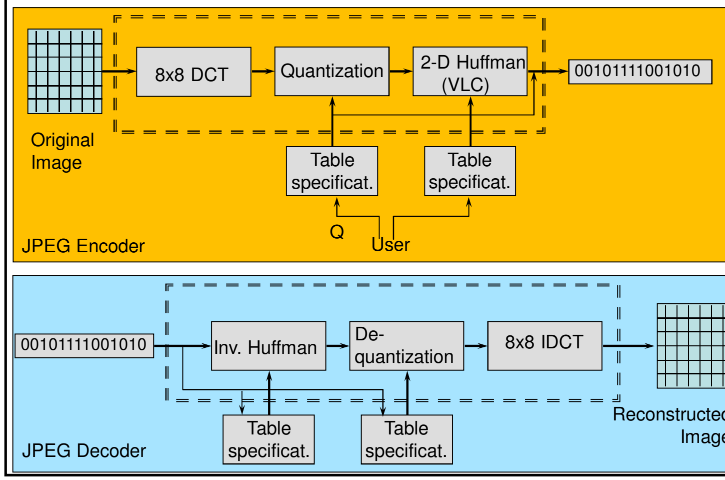
JPEG Requirements

- **Usability:**
  - Wide range of bit rates and qualities
  - Moderate complexity
  - Encoder/decoder of comparable complexity
  - Sequential image build up (top-left to bottom-right)
- **Quality guidelines:**
  - 0.25-0.50 bit/color pixel: Moderate to good quality
  - 0.50-0.75 bit/color pixel: Good to very good quality
  - 0.75-1.50 bit/color pixel: Excellent quality
  - >1.50 bit/color pixel: Indistinguishable from original

JPEG Coding Modes

- **Baseline JPEG ("sequential mode")**
  - 8x8 DCT based compression
  - Quantization: normalization (=weight) matrix and rounding
  - DC coefficients are DPCM coded (lossless)
  - AC coefficients are {zero run, amplitude} Huffman coded
  - Up to 4 color components and 4 different normalization matrices
  - Bit stream can store normalization matrices and VLC table
  - No explicit bit rate control
- **Progressive mode (currently: *Internet* pictures)**
- **Hierarchical mode**
- **Lossless compression mode: JPEG-LS**

JPEG Coding System Architecture



JPEG DCT Weighting & Quantization

- **Quantization:**

$$\hat{y}(u,v) = Q[y(u,v)] = \text{round}\left(\frac{y(u,v)}{Q' W(u,v)}\right)$$

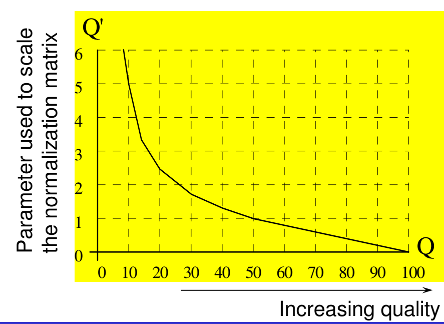
- **Recommended JPEG normalization matrix**

N(u,v)=W(u,v)

$$W(u,v)=\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

JPEG User Controllable Quality

- **User has control over a "quality parameter" that runs from 100 ("*perfect*") to 0 ("*extremely poor*")**



JPEG 8x8 DCT Example – (1)

$$x(k,l)=\begin{bmatrix}139&144&149&153&155&155&155&155\\144&151&153&156&159&156&156&156\\150&155&160&163&158&156&156&156\\159&161&162&160&160&159&159&159\\159&160&161&162&162&155&155&155\\161&161&161&161&160&157&157&157\\162&162&161&163&162&157&157&157\\162&162&161&161&163&158&158&158\end{bmatrix}$$

DCT transform is exactly defined in JPEG standard

average x 8

$$y(u,v)=\begin{bmatrix}1260&-1&-12&-5&2&-2&3&1\\-23&-17&-6&-3&-3&0&0&-1\\-11&-9&-2&2&0&-1&-1&0\\-7&-2&0&1&1&0&0&0\\-1&-1&1&2&0&-1&1&1\\2&0&2&0&-1&1&1&-1\\-1&0&0&-1&0&2&1&-1\\-3&2&-4&-2&2&1&-1&0\end{bmatrix}$$

JPEG 8x8 Block Coding, Example – (2)

$$\hat{y}(u,v)=\begin{bmatrix}79&0&-1&0&0&0&0&0\\-2&-1&0&0&0&0&0&0\\-1&-1&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\end{bmatrix}$$

Quantization using Q' = 1

- ***DC***: **Difference with quantized DC coefficient of previous block is Huffman encoded**
- ***AC***: **Zigzag scanning of coefficients, and convert to (*zero run-length, amplitude*) combinations:**
  - Input : (79) 0 -2 -1 -1 -1 0 0 -1 0 0 0 …
  - Output: Code(79) {1,-2} {0,-1} {0,-1} {0,-1} {2,-1} EOB

JPEG VLC Coding of AC Coefficients

- **1. Amplitudes are classified in categories**

Category	*AC Coefficient Range*
1	-1,1
2	-3,-2,2,3
3	-7,....,-4,4,....,7
4	-15,....,-8,8,....,15
5	-31,....,-16,16,....,31
6	-63,....,-32,32,....,63
7	-127,....,-64,64,....,127
8	-255,....,-128,128,....,255
9	-511,....,-256,256,....,511
10	-1023,....,-512,512,....,1023

- **2. Events (*zero run-length, categories*)→ Huffman coded**
- **3. Sign and amplitude offset within a category are FLC coded (required #bits = category number)**

JPEG AC Coefficient Huffman Coding Table (run, category)

Zero Run	*Category*	*Code length*	*Codeword*
0	1	2	00
0	2	2	01
0	3	3	100
0	4	4	1011
0	5	5	11010
0	6	6	111000
0	7	7	1111000
.	.	.	.
.	.	.	
1	1	4	1100
1	2	6	111001
1	3	7	1111001
1	4	9	111110110
.	.	.	.
.	.	.	.
2	1	5	11011
2	2	8	11111000
.	.	.	.
.	.	.	.
3	1	6	111010
3	2	9	111110111
.	.	.	.
.	.	.	.
4	1	6	111011
5	1	7	1111010
6	1	7	1111011
7	1	8	11111001
8	1	8	11111010
9	1	9	111111000
10	1	9	111111001
11	1	9	111111010
.	.	.	.
.	.	.	.
EOB		4	1010

Breakdown of coding

- signed amplitude = **category + offset + sign**

JPEG VLC Coding Example – (3)

- **The event series**
  {1,-2} {0,-1} {0,-1} {0,-1} {2,-1}   EOB
  now becomes as bit string

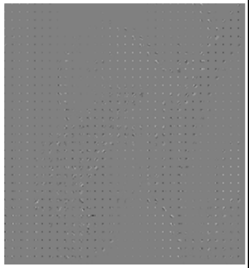
  111001 01 / 00 0 / 00 0 / 00 0 / 11011 0 / 1010

- **Bit rate for AC coefficients in this DCT block**
  27 bits/64 pixels = 0.42 bit/pixel

JPEG VLC Coding Example – (4)

Coded at Q=10

Quantization errors



Coded DCT block

JPEG Customization

- **Design of the normalization matrix** $N(u,v)=W(u,v)$
- **Design of the Huffman tables**
  - Usually done by quantization of all picture data,
  - Followed by a histogram analysis of the (zero run-length, amplitude) combinations
- **Rate-control mechanism (trial and error)**
- **No possibility to vary Δ over DCT blocks (global Q)**
- **Suboptimal alternative is to locally discard non-zero quantized DCT coefficients if they "cost" too much**
  - Long zero run followed by small non-zero amplitude

JPEG follow-up: JPEG 2000

- **Recent update of existing standard**
- **Advanced wavelet coding technique**
  - Wavelet analysis of picture with filter banks
  - Wavelet functions for suited for scalable video
- **Special band-coding algorithms**
  - Parent-children coding technique of (non-)zeros
  - Special quantization and VLC
- **Standard is accepted in digital Cinema, prof. imaging**

17

5LSE0 - Mod 08
Part 2
Intraframe DCT compression for digital video camcording: DV standard

18

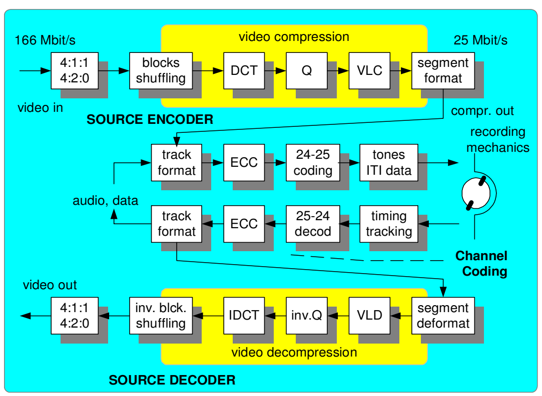
Design considerations of DV compression

Contents

- System constraints
- Cassette size and tape
- Video mapping on sync blocks
- Video compression
- Data shuffling for trick play
- Conclusions

19

System constraints / DV architecture

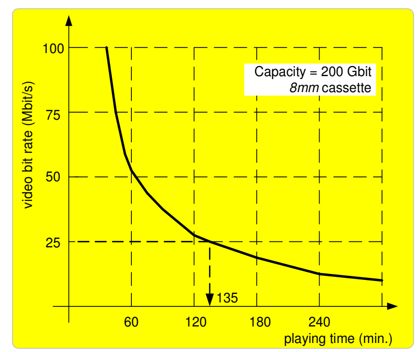


20

System / Cassette size vs. playing time

Assumptions

- ME tape
- track width 10 µm
- bit length 0.25 µm
- 60% of data video
- video compression
- 25 Mbit/s

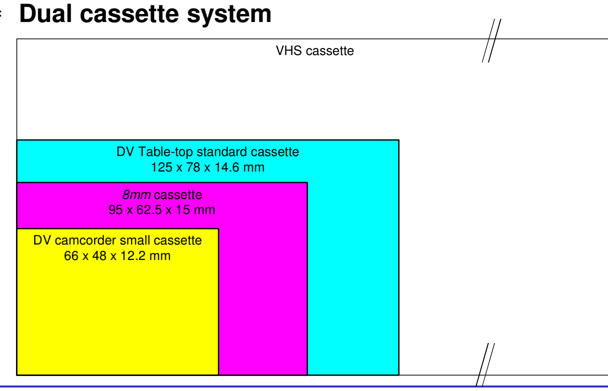


System / Cassette system outline – (1)

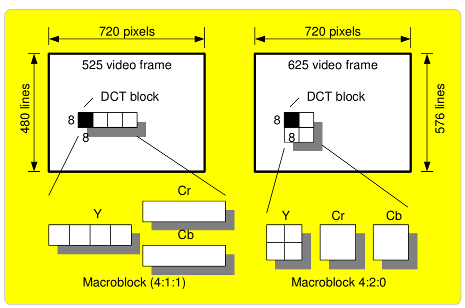
- **Pocketable Camcorders**
  - more compact than analogue *8mm*
  - playing time less important
- **Home recording of HDTV**
  - minimum playing time 135 min. for movies
- **Home recording of Standard Definition TV**
  - ratio HDTV to SDTV of 2:1
  - minimum playing time 270 min. for SDTV
- **One-cassette system**

System / Cassette system outline – (2)

- **Dual cassette system**

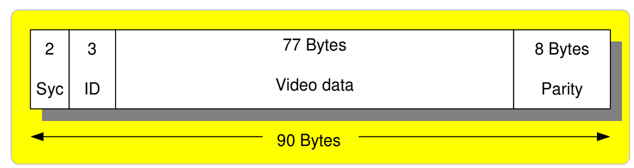


System / Video Mapping Sync Blocks – (1)



- **1350 / 1620 Macroblocks per frame: map on 10 / 12 tracks / frame and 70 - 140 Sync Blocks / track**

System / Mapping Sync Blocks – (2)

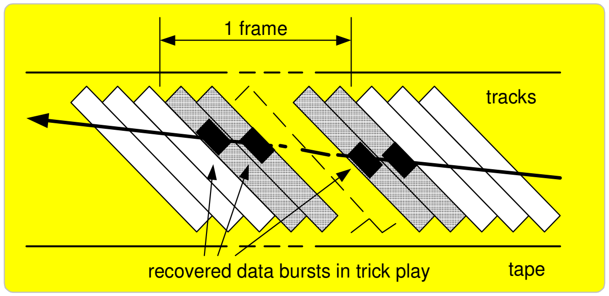


- **Sync block is data packet in recording channel**
- **Relatively short length of 90 Bytes**
- **System aspects: robustness, search speed**

System / Recording system constraints - (1)

- Editing: preferably on picture basis
- Repeated (de-)compression: robust for dubbing
- Multitrack format: set of multiple tracks for 1 frame
- Forward and backward search
  - picture recovery from small data bursts
- Robustness: under all circumstances
- High picture quality: beyond analog formats

System / Recording system constr'ts – (2)

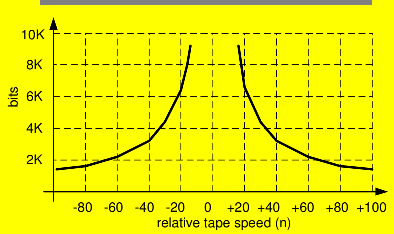


- Segmented recording and recovery

System / Mapping Sync Blocks – (3)

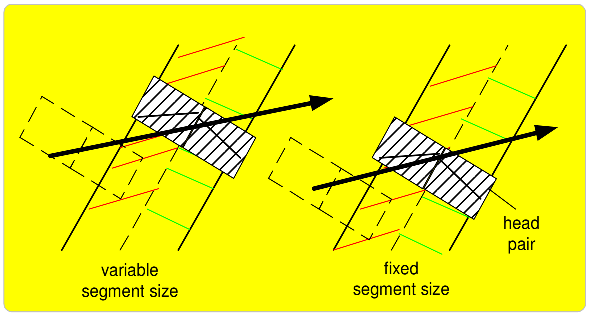
- 10 / 12 tracks per video frame (one channel rate)
- 135 Macroblocks per track
- 5 Macroblocks / video segment: division of 27

MBs per segment	SBs per segment	Video SB/track	Total SB per track	n_max speed
5	3	81	108	54
5	4	108	144	72
5	5	135	180	90



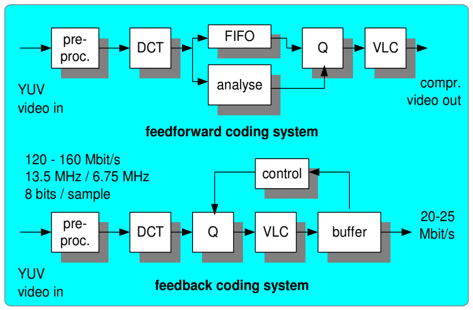
System / Fixed or variable segments

Note the predictive position of heads at specific tape speeds



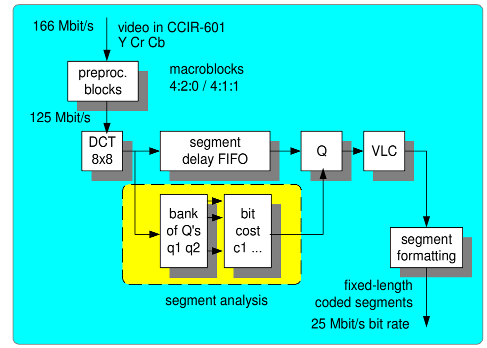
System / Feedback or feedforward system

* MPEG is feedback system
* Fixed segment size coding requires other architecture



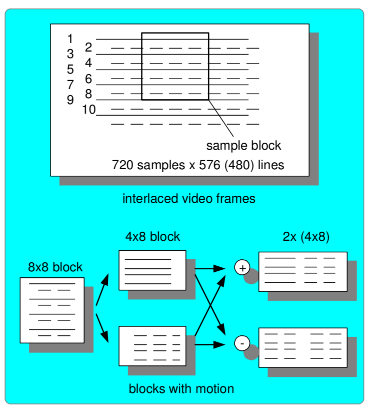
Video compression / Feedforward coding system

* Independent coding of segments
* Fixed-length compression for segments
* Results in small accessible units



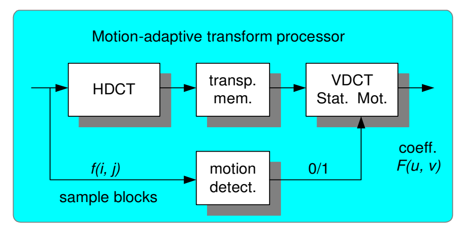
Video coding / Motion-adaptive DCT – (1)

* Video field interlacing in normal mode
* Unequal time sampling
* Sort block data by same time instant in case of motion
* Separate LF and HF temp data



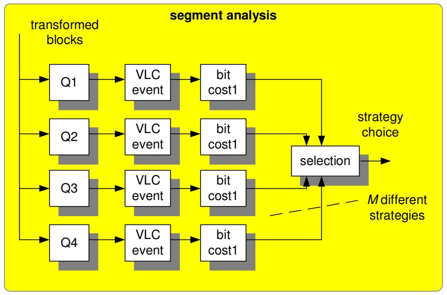
Video coding / Motion-adaptive DCT – (2)

* Motion detection for *a-priori* decision
* Split horizontal and vertical proc. of transform
* Vertical DCT should be 8-point (static) or 2x4-point (motion)



Video compr. / Segment analysis – (1)

- All K blocks of segment are analysed upfront
- In hardware: various quantizers in parallel
- Preliminary quantization and event coding for bit cost computation



Video compr. / Segment analysis – (2)

Problem statement

- For all blocks k with $0 \le k \le K-1$ do
- Determine for strategy m and k-th block the bit cost

$$C_{m,k} = Cc_{m,k} \sum_{u,v} [Q_{m,k}(F_k(u,v))]$$

- Perform all m quantizer strategies for $1 \le m \le M$

$$R_m = \sum_{k=0}^{K-1} C_{m,k} \,/\, KN^2$$

- Select best strategy m_{opt} with $0 \le m_{opt} \le M-1$ such that the deviation of the desired rate is minimal, thus

$$| R_m - R_d | \cong 0$$

Video compr. / Quantization – (1)

Quantization of NxN DCT blocks

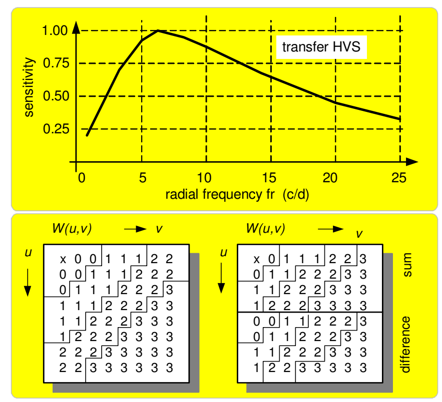
(1). **Frequency-dependent:** weighting

(2). **Adaptive to local image statistics**

(3). **Uniform block quantization**

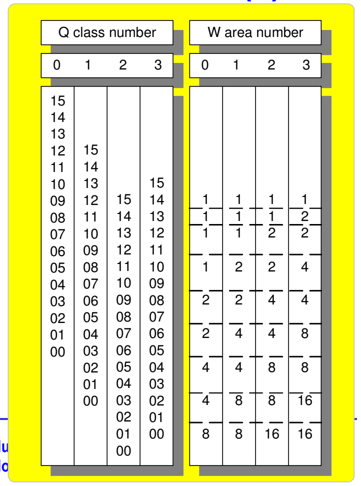
Video compr. / Quantization – (2)

- **Weighting is frequency-dependent**
- **Relates to Human Visual System**
- **DV has strongly discretized weighting (only 4 weights)**



Video compr. / Quantization – (3)

- **Local adaptivity**
  - 4 classes sufficient
  - any metric allowed, e.g. SUM $F^2(u,v)$
- **Uniform quantization**
  - division by step size S
  - combine $W(u, v) / S$ into powers of 2



Video compr. / Var.-Length coding – (1)

Algorithm: Coding **{runlength, amplitude}** combinations

(1). Count zeros until non-zero coefficient occurs = runlength

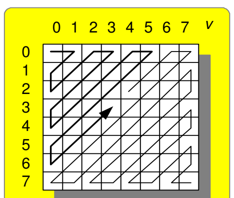
(2). Combine actual runlength with non-zero coefficient into one statistical *event*

(3). Apply one single codeword for the event

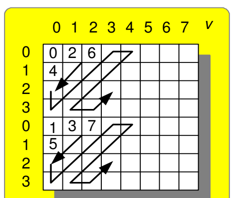
Example: DC +34 0 -16 +11 0 0 +2 -1 0 0 0 0 …, becomes DC (0,+34) (1,-16) (0, +11) (2,+2) (0,-1) EOB

Video compr. / Var.-Length coding – (2)

- **Algorithm Step 1: Scanning of coefficients**
  - purpose is zero clustering
  - start with relevant info and conclude with EOB
  - Adaptive to motion mode (2 x (4x8) subblocks)



zigzag scanning in **static mode**



muxed scanning in **motion mode**

Video compr. / Var.-Length coding – (3)

- Algorithm Step 2: **Runlength count and events**
- Algorithm Step 3**: Coding of {runlength, amplitudes} events**
  - 2-Dimensional coding table (words + wordlengths)
  - Entries are runlength and non-zero amplitude
  - Exceptions: concatenation of existing codewords, e.g. (5,+7) = (4,0) + (0,+7)

Video compr. / Var.-length coding – (4)

* Two-dimensional coding table of wordlengths

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

Video compr. / Var.-Length coding – (5)

* Two-dimensional table of codewords

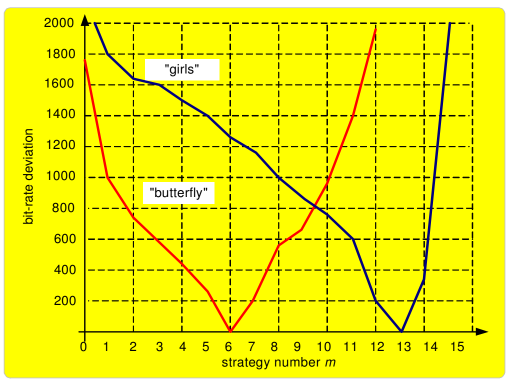
code	runlength	amplitude
0110	EOB	
00s	0	1
010s	0	2
0111s	1	1
1000s	0	3
1001s	0	4
1010 0s	2	1
1010 1s	1	2
1011 0s	0	5
1011 1s	0	6
1100 00s	3	1
1100 01s	4	1
1100 10s	0	7
1100 11s	0	8
1101 000s	5	1
1101 001s	6	1
1101 010s	2	2
1101 011s	1	3
1101 100s	1	4
1101 101s	0	9
1101 110s	0	10
1101 111s	0	11

code	runlength	amplitude
11100 000s	7	1
11100 001s	8	1
11100 010s	9	1
11100 011s	10	1
11100 100s	3	2
11100 101s	4	2
11100 110s	2	3
11100 111s	1	5
11101 000s	1	6
11101 001s	1	7
11101 010s	0	12
11101 011s	0	13
11101 100s	0	14
11101 101s	0	15
11101 110s	0	16
11101 111s	0	17
11110 0000s	11	1
11110 0001s	12	1
11110 0010s	13	1
11110 0011s	14	1

code	runlength	amplitude
11110 0100s	5	2
11110 0101s	6	2
11110 0110s	3	3
11110 0111s	4	3
11110 1000s	2	4
11110 1001s	2	5
11110 1010s	1	8
11110 1011s	0	18
11110 1100s	0	19
11110 1101s	0	20
11110 1110s	0	21
11110 1111s	0	22
1111 1000 00s	5	3
1111 1000 01s	3	4
1111 1000 10s	3	5
1111 1000 11s	2	6
1111 1001 00s	1	9
1111 1001 01s	1	10
1111 1001 10s	1	11
1111 1001 110	esc (0,0)	
1111 1001 111	esc (1,0)	
1111 1010 1100	esc (2,0)	

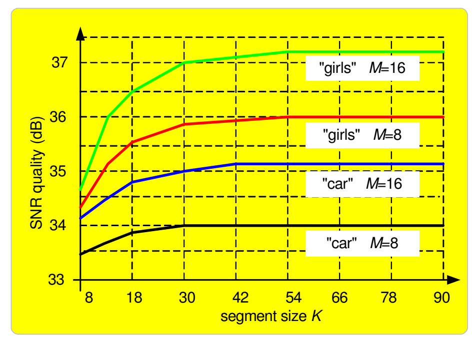
Video compr. / System optimization – (1)

* **Strategy vs. bit rate**
  - 16 different strateg's
  - Variable picture material



Video compr. / System optimization – (2)

* **Segment size in DCT blocks**
  - Optimal choice segment size K
  - Number of strategies M
  - CCIR-601, 4:2:0, 25 Mbit/s bit rate
  - Result: K = 30, M=16



45

Video compr. / Robust MB format – (1)

Ultra-Robust segment and MB format

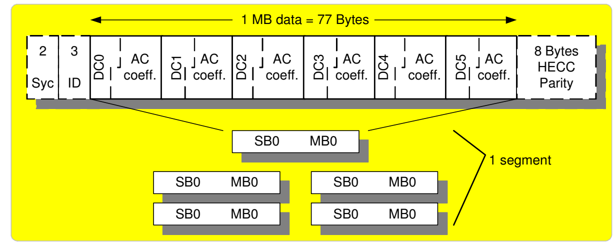
* Format for *compressed* data
* Enables high search speeds
* Robust for residual errors, because of partial ECC
* Key aspect for portable equipment

46

Video compr. / Robust MB format – (2)

Principal elements of format

(1). Fixed-length coded segments
(2). One MB (macroblock) per sync block
(3). Special low-freq. part of DCT blocks on fixed positions



47

5LSE0 - Mod 08
Part 3

Intraframe DCT compression for digital video camcording: Fast Searching in the DV standard

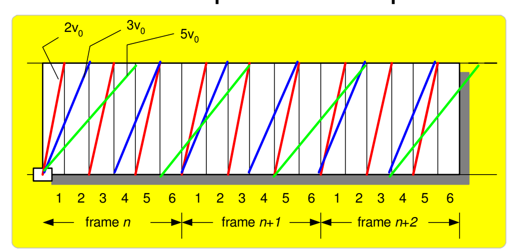
(Addendum to Part 2)

48

Video compr. / Searching at speed

Picture search at different speeds

(1). Aim at full-image periodic refresh
(2). Not every tape speed is allowed
(3). Not a divisor or multiple of track count per frame

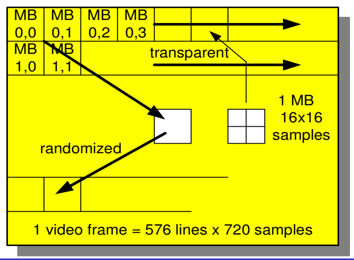


Video compr. / Video shuffling – (1)

Video shuffling to support picture search

(1). Transparent (process as received): minimum memory

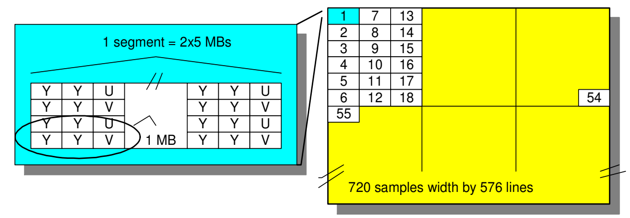
(2). Randomized: spreads moving and non-moving areas



Video compr. / Video shuffling – (2)

(3). Super MacroBlock Construction (SMBC):

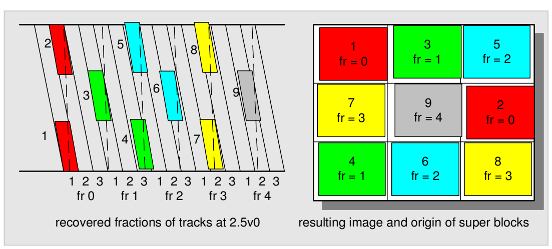
- ∗ Image divided in nearly square super blocks
- ∗ Super blocks segmented into smaller super blocks
- ∗ Coinciding super blocks fit as a fraction to tape speeds



Video compr. / Video shuffling – (3)

(3). Super MacroBlock Construction (SMBC):

- ∗ rely on recovered fraction 1/(2(n-1)) of track at nv_0 speed
- ∗ construct larger blocks of smaller blocks fitting with fractions (consider example at 2.5 v_0 with 3 double tracks per frame)



Video compr. / Video shuffling – (4)

(1). Transparent (Example at 2.5 v_0 with 3 double tracks per frame)

Video compr. / Video shuffling – (5)

(2). Randomized (Example 2.5 v_0 with 3 double tracks per frame)

Video compr. / Video shuffling – (6)

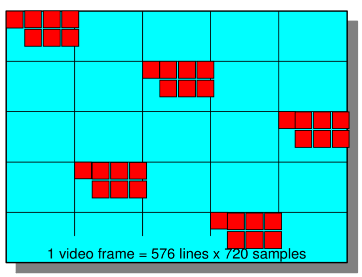
(3). SMBC (Example 2.5 v_0 with 3 double tracks per frame)

Video compr. / Video shuffling – (7)

Finally adopted frame-based shuffling

* compromise on randomized (global) and SMBC (local)

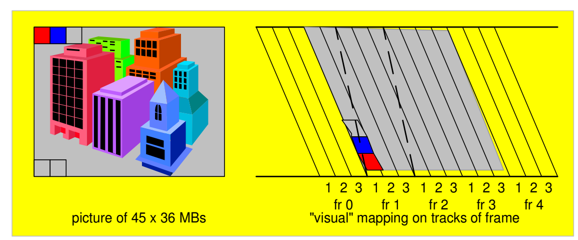
- Gradual filling of image area in clusters of MBs
- Affects both coding efficiency and picture search quality



Video compr. / Video shuffling – (8)

Additional shuffling step for trick play

* Performed after compression: no influence on normal PQ
* Lower speeds require larger adjacent areas
* Coherent mapping of image required



DV Video Coding / Conclusions – (1)

* **(1) Cassette size and tape**
  – Minimized for camcording
  – ME tape and 2.5 sq.micron / bit density
  – Intraframe DCT Video compression to 25 Mbit/s
  – Sufficient playing time
* **(2) Small segment feedforward video coding**
  – Independent compression of each segment
  – Motion-adaptive for interlace
  – Low-cost intraframe coding
  – High accessibility for editing
  – High robustness

DV Video Coding / Conclusions - (2)

* **(3) Special formatting and shuffling**
  – Distributed MBs for optimal compression
  – Coherent image areas on tape
  – Small segment size for high-speed search
  – Very robust inner segment format (even without ECC)

DV is a clear technological and market success, the compression has been adopted even for Hi8, and professional applications!