SMPTE Journal, June 2000

PROPOSED SMPTE STANDARD

SMPTE 334M

for Television —
Vertical Ancillary Data Mapping for Bit-Serial Interface

1 Scope

This standard defines a method of coding which allows data services to be carried in the vertical ancillary data space of a bit-serial component television signal conforming with SMPTE 292M or ANSI/SMPTE 259M.

This includes data broadcast services intended for the public as well as broadcaster internal control and communications.

Despite the reference to the bit-serial interface, nothing in this specification precludes its use in a parallel digital interface for component digital video signals.

The data described in this standard may also be transported in KLV format according to SMPTE 336M, or via other means.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent edition of the standards indicated below.

ANSI/SMPTE 125M-1995, Television — Component Video Signal 4:2:2 — Bit-Parallel Digital Interface

ANSI/SMPTE 259M-1997, Television — 10-Bit 4:2:2 Component and $4f_{sc}$ Composite Digital Signals — Serial Digital Interface

SMPTE 291M-1998, Television — Ancillary Data Packet and Space Formatting

SMPTE 292M-1998, Television — Bit-Serial Digital Interface for High-Definition Television Systems

ANSI/EIA 608-1994, Recommended Practice for Line 21 Data Service

EIA 708-B, Digital Television Closed-Captioning (DTVCC)

EIA 766, U.S. Region Rating Table (RRT) and Content Advisory Descriptor for Transport of Content Advisory Information Using ATSC A/65 Program and System Information Protocol (PSIP)

3 Location of vertical ancillary data

The data packets are located in the active line portion of one or more lines in the vertical ancillary space. Data may be located in any lines in the area from the second line after the line specified for switching to the last line before active video, inclusively.

Individual data services are not assigned to any specific data lines; receiving equipment should identify and select services on the basis of their ANC DID and SDID fields.

Because ANC data may be located in the lines immediately preceding active video, manufacturers of video compression equipment should ensure that these data bits are not included in video compression calculations.

The chrominance (C_b/C_r) and luminance (Y) data are carried in two separate streams within the 292M signal, complete with their own ANC data flags and CRCs.

THIS PROPOSAL IS PUBLISHED FOR COMMENT ONLY

Defined data services shall be carried in the Y stream. Other data services may be inserted into either one of these streams without restrictions.

In the 259M/125M signal, the chrominance and luminance data are carried in a single stream. In this case, all data services shall be carried in this stream with a single ANC data flag and CRC.

4 Format of VANC data packets

Each data packet follows the format defined in SMPTE 291M for a type 2 ANC packet. It consists of the ancillary data flag (ADF), the data ID (DID), the secondary data ID (SDID), the data count (DC), the user data words (UDW), and the checksum (CS). The UDW consists of the data payload plus forward error correction overhead.

4.1 ANC packet header format

The ADF has the value 000_h $3FF_h$ $3FF_h$.

The following value of DID is used for the closed-captioning services defined in this standard: 161_h (61_h plus parity bits per 291M). A second value of DID (162_h = 62_h plus parity) is used for other services which are identified by this standard, and whose format is specified by recommended practices.

Other data services which are internal to a broadcast network may use DID values in the ranges specified for user application data (40_h-$5F_h$ and $C0_h$-DF_h). These DID values are not registered.

The specified values of DID (61_h and 62_h) identify type 2 ANC packets. In each packet, the SDID code identifies the type of data. Table 1 shows the values of DID and SDID for services defined within this standard. Table 2 shows the values of DID and SDID for other services.

DC is a count of the number of words in the UDW; bits b7-b0 of DC represent the number of words of user data; bits b8 and b9 are parity per 291M.

For defined services such as captioning, the format of the data in the UDW is defined in this specification or in a normative reference. For other data services, the data content is not specified here, and the value of DC is variable.

4.2 UDW format

All data services consist of 8-bit data bytes, which are transmitted in bits b7-b0 of the 10-bit data word. Bit b8 is even parity for b7 through b0, and b9 = not b8. In addition to providing a simple error detection capability, this avoids transmitting data which match one of the code words 0-3 and 1020-1023 which are prohibited by SMPTE 292M and ANSI/SMPTE125M.

Table 1 – Defined data services

Service	DID	SDID	DC
Closed captioning (EIA 708-B)	61_h (161_h)	1 (101_h)	Variable
EIA 608 data	61_h (161_h)	2 (102_h)	3 (203_h)

Table 2 – Variable-format data services

Service	DID	SDID	DC
Program description (DTV)	62_h (162_h)	1 (101_h)	Variable
Data broadcast (DTV)	62_h (162_h)	2 (102_h)	Variable
VBI data	62_h (162_h)	3 (203_h)	Variable

The data payload for each service is inserted into the UDW of the ANC packet as a sequence of 10-bit words. The number of words is indicated in the DC field of the ANC packet header.

4.3 Defined services

The services shown in table 1 have their format defined in this clause. The values in parentheses for DID, SDID, and DC include parity bits b8 and b9.

4.3.1 Format of the closed captioning (EIA 708-B) packet

The payload of the closed captioning (EIA 708-B) packet is the caption distribution packet (CDP) defined in EIA 708-B. This packet has a variable length.

4.3.2 Format of the ANC EIA 608 (VBI) packet

In NTSC video, the closed captioning, content advisory, and other services are carried in a format defined by the EIA 608 standard. Closed captioning may be carried in line 21 of either field. Content advisory and other data may be in line 21 of field 2 only.

These can be carried in an ANC packet in a 292M stream to allow the EIA 608 data waveform to be recreated and reinserted into an NTSC signal produced by converting the DTV signal into an analog signal at a station. The format of this ANC packet is defined in annex A.

4.4 Other data services

Table 2 lists other data services whose format is not specified by this standard. Their DID and SDID values are specified here to ensure that they can be correctly and consistently recognized and routed.

The DTV program description service carries data which pertain to the video and audio programs. Its contents are defined in the forthcoming SMPTE RP 207.

The DTV data broadcast service carries data intended for broadcast to the public along with the video and audio programs. Its contents are the subject of a future recommended practice.

The VBI data service is intended for use in reconstituting data in the VBI of a standard-definition analog video signal produced from the digital video program. Its contents are defined in the forthcoming SMPTE RP 208.

5 Timing of data and video

There is no specific provision in this standard for ensuring that the relative timing between the video and its embedded VANC data is correct. The only timing relationship that exists is created when the data are embedded in the video. Once that relationship is established, the deterministic nature of 292M or 259M transport ensures that the relationship is preserved.

Annex A (normative)
Format of the ANC EIA 608 packet

The data payload for EIA 608 data is 2 bytes per line. The ANC packet encapsulates these two bytes without modification, and adds a byte which identifies the VBI line and field to be used for insertion. The data count (DC) is therefore 3 (203_h).

The format of the packet is as follows:

Header:
- ADF (3 words)
- DID = 161_h
- SDID = 102_h
- DC = 203_h

UDW:
- LINE (1 word)
- EIA 608 data (2 words)

Suffix:
- CS (1 word)

The LINE value at the start of the UDW represents the field number and VBI line where the data are intended to be carried. Bit b7 of the LINE value is the field number (0 for field 2; 1 for field 1). Bits b6 and b5 are 0. Bits b4-b0 form a 5-bit unsigned integer which represents the offset (in lines) of the data insertion line, relative to the base VBI frameline (line 9 of 525-line field 1, line 272 of 525-line field 2, line 5 of 625-line field 1, line 318 of 625-line field 2).

Annex B (informative)
Bibliography

ANSI/SMPTE 296M-1997, Television — 1280×720 Scanning, Analog and Digital Representation and Analog Interface

ANSI/SMPTE 299M-1997, Television — 24-Bit Digital Audio Format for HDTV Bit-Serial Interface

ATSC A/53, ATSC Digital Television Standard

ATSC A/65, Program and System Information Protocol for Terrestrial Broadcast and Cable

SMPTE 274M-1998, Television — 1920 × 1080 Scanning and Analog and Parallel Digital Interfaces for Multiple Picture Rates

Forthcoming SMPTE 336M, Television — Data Encoding Protocol Using Key-Length-Value

Forthcoming SMPTE RP 207, Transport of Program Description Data in Ancillary Data Packets

Forthcoming SMPTE RP 208, Transport of VBI Packet Data in Ancillary Data Packets

PROPOSED SMPTE STANDARD

SMPTE 342M

for Television — Encoding Process and Data Format for HD-D5 Compressed Video — 1080i and 720p Systems

1 Scope

This standard defines the encoding process of the HD-D5 video compression and its data format for the 1080i/59.94 system (hereafter referred to as the 1080i system) and the 720p/59.94 system (hereafter referred to as the 720p system).

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent edition of the standards indicated below:

ANSI/SMPTE 296M-1997, Television — 1280 × 720 Scanning, Analog and Digital Representation and Analog Interface

SMPTE 274M-1998, Television — 1920 × 1080 Scanning and Analog and Parallel Digital Interfaces for Multiple Picture Rates

3 Acronyms

BUF	Buffer memory
C	Color-difference signal
C3RMB	Compressed data of 3 RMBs
C(t, u)	The value of the DCT coefficient at frequency (t, u)
C_B/C_R	Color-difference signal
CC0~CC2	Categories for C DCT block
Ccoef( )	C DCT CG
CG	Coefficient group
CGNR	CG number of one Y/C DCT coefficient block in one RMB
CGNS	CG number of one Y/C DCT coefficient block in one SMB
CN	C3RMB number in one RMBG
CRcoef ( )	Rearranged C DCT CG
CS	C DCT block number in one SMB
CY0~CY3	Categories for Y DCT block
DCT	Discrete cosine transform
DIF	Digital interface
DIF (n)	DIF block numbered n
DN	DIF block number
EOB	End of block code
EOM	End of 3 RMBs code
exnor	Logical exclusive nor
f( )	Offset value table for SMBG distribution
FCB	Category flag of C_B DCT block
FCB'	Category flag of C_B DCT block
FCR	Category flag of C_R DCT block
FCR'	Category flag of C_R DCT block
FFL	Field number flag
FMB	Category flag of the MB
FMB'	Category flag of the MB
FYa ~ FYd	Category flags of the four DCT blocks (Ya ~ Yd) of the MB
FYa' ~ FYd'	Category flags of the four DCT blocks (Ya ~ Yd) of the MB
H	The horizontal SMB position number in one video field (1080i system) or one video frame (720p system)
HR	The column position number of RMB
HS	The column position number of SMB in one SMBG
IDCT	Inverse discrete cosine transform
int (A)	Integer part of A
LEN	The byte length of C3RMB
MB	Macro block
mod	Modulus operator
N.A.	Not applicable
Offset ( )	Offset value for RMB shuffling
P (r, s)	The value of the pixel at the position (r, s) in Y/C DCT block
Qno	Quantization number
Qstep	Quantization step value
r	The horizontal pixel position number in Y/C DCT block
Rg	The RMBG number within the RMBs
RMB	Rearranged macro block
RMBG	Rearranged macro block group
Rn	The number of RMB coding order in each RMBG
s	The vertical pixel position number in Y/C DCT block
SA	The starting address of the remainder data in buffer memory
SABM	One byte data of SA (two bytes)
Sg	The SMBG number in one video field (1080i system) or one video frame (720p system)
SMB	Super macro block
SMBG	Super macro block group
t	The horizontal frequency number in Y/C DCT coefficient block
Table CY0~3	Set up value tables for Y weighting function
Table CC0~2	Set up value tables for C weighting function
u	The vertical frequency number in Y/C DCT coefficient block
V	The vertical position number of SMB in one video field (1080i system) or one video frame (720p system)
VLC	Variable length coding
VR	The row position number of RMB
VS	The row position number of SMB in one SMBG
W (t, u)	Weighting value at frequency (t, u)
Y	Luminance signal
Ya ~ Yd	Four Y DCT blocks in one MB
Ycoef ( )	Y DCT CG
YR	The Y DCT coefficient block number in one RMB
YRcoef ( )	Rearranged Y DCT CG
YS	The Y DCT block number in one SMB
Z	The row position number of the RMB after RMB shuffling
ZRL	Code of 15 successive zero coefficients followed by a coefficient of zero amplitude

THIS PROPOSAL IS PUBLISHED FOR COMMENT ONLY

4 Video processing

4.1 Overview

Luminance (Y) and color-difference components (C_B and C_R) from 1080i or 720p video signal are sampled by 74.25/1.001 MHz and 37.125/1.001 MHz, respectively.

After discarding samples in vertical and horizontal blanking periods, active video samples are divided into four super macro block groups (SMBG) per field (1080i) or per frame (720p). Each SMBG consists of 1080 super macro blocks (SMB).

Each SMB consists of two MBs. Each MB consists of four luminance DCT blocks (8 × 4 pixel matrix) and one each of C_B DCT block (8 × 8 pixel matrix) and C_R DCT block (8 × 8 pixel matrix).

As described later, two horizontally adjacent luminance DCT blocks are overlapped by one pixel column at their junction. Two horizontally adjacent chrominance DCT blocks are overlapped by one pixel column at their junction when they are formed into an SMB.

Each DCT block is transformed to represent DC and AC coefficients. Coefficients are weighted through the prearranged categories prior to shuffling, then formed into rearranged MBs (RMB).

DCT coefficients within one rearranged MB group (RMBG) are quantized, and made into a fixed length data set through VLC.

The VLC output code words from one RMBG are formed into 360 DIF blocks.

The compressed video data for one 1080i field or one 720p frame consist of 5760 DIF blocks.

The block diagram of the outline about video processing is shown in figure 1.

4.2 Video signal

4.2.1 Sampling process

The sampling structure is defined in SMPTE 274M and ANSI/SMPTE 296M. Sampling structures of the luminance (Y) and the two color-difference signals (C_B/C_R) are described in table 1.

4.2.1.1 Line structure in one field (1080i system) or frame (720p system)

For the 1080i system, 540 lines for Y, C_B, and C_R signals from each field shall be transmitted. For the 720p system, 720 lines for Y, C_R, and C_B signals from each frame shall be transmitted. The transmitting lines on a television frame are defined in table 1.

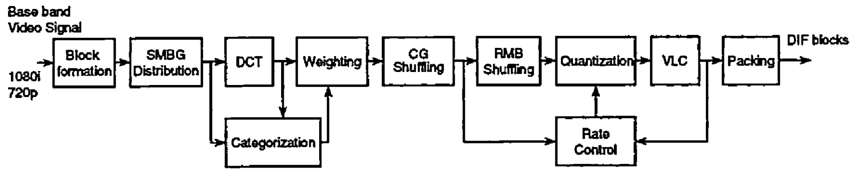

Figure 1 – Block diagram of outline about video processing

Table 1 – Construction of video signal sampling

		1080i system		720p system	
Sampling frequency	Y	74.25 MHz / 1.001			
	C_B/C_R	37.125 MHz / 1.001			
Total number of pixels per line	Y	2200		1650	
	C_B/C_R	1100		825	
Number of active pixels per line	Y	1920		1280	
	C_B/C_R	960		640	
Total number of lines per frame		1125		750	
Number of active lines per frame		1080		720	
Active line numbers		Field 1	21 to 560	Frame	26 to 745
		Field 2	584 to 1123		
Quantization		Each sample is linearly quantized to 10 bits for Y, C_B, and C_R			
The relation between video signal level and quantized level	Scale	4 to 1019			
	Y	Quantized level: Video signal level of white: Video signal level of black:	877 940 64		
	C_B/C_R	Quantized level: Video signal level of gray:	897 512		

4.2.1.2 Pixel structure in one field (1080i) / in one frame (720p)

– 1080i system: All sampled pixels, 1920 luminance pixels per line and 960 color-difference pixels, are retained for processing as shown in figure 2. The sampling process starts simultaneously for both luminance and color-difference signals.

– 720p system: All sampled pixels, 1280 pixels per line and 640 color-difference pixels, are retained for processing as shown in figure 3. Sampling processes start simultaneously for both luminance and color-difference signals.

4.3 Block formation

4.3.1 DCT block, macro block (MB), and super macro block (SMB)

4.3.1.1 DCT block

The Y pixels in a field (1080i system) and in a frame (720p system) shall be divided into rectangular areas of 15 horizontal pixels and 4 lines. Two Y DCT blocks (one Y DCT block pair) are made from each one of the rectangular areas as shown in figure 4. In each Y DCT block pair, the rightmost pixel in the left DCT block is overlapped with the leftmost pixel in the right DCT block (overlapped blocking).

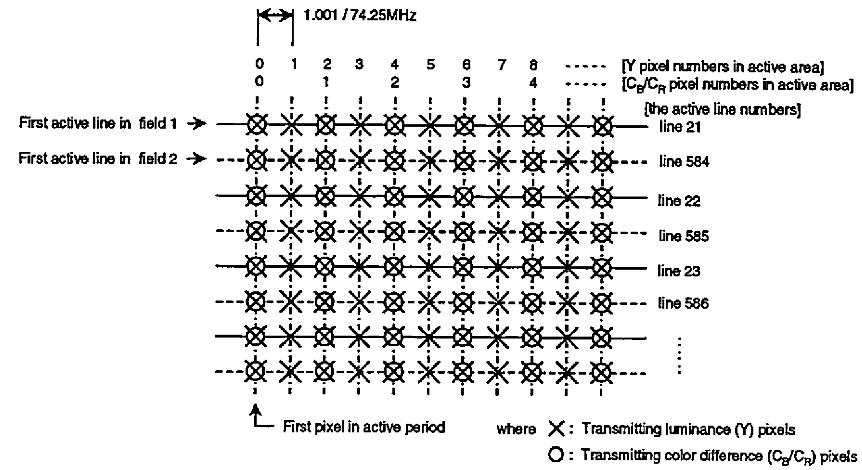

Figure 2 – Transmitting samples of 1080i system

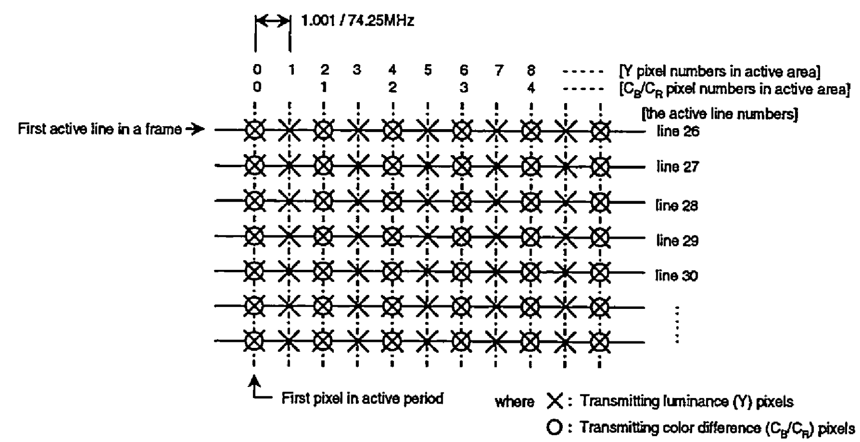

Figure 3 – Transmitting samples of 720p system

The C_B/C_R pixels in a field (1080i) and in a frame (720p) shall be divided into rectangular areas of 15 horizontal pixels and 8 lines. Two C DCT blocks (one C DCT block pair) are made from each one of the rectangular areas as shown in figure 5. In each C DCT block pair, the rightmost pixel in the left block is overlapped with the leftmost pixel in the right block (overlapped blocking). Overlapped blocking is used for the robustness of error (see annex A).

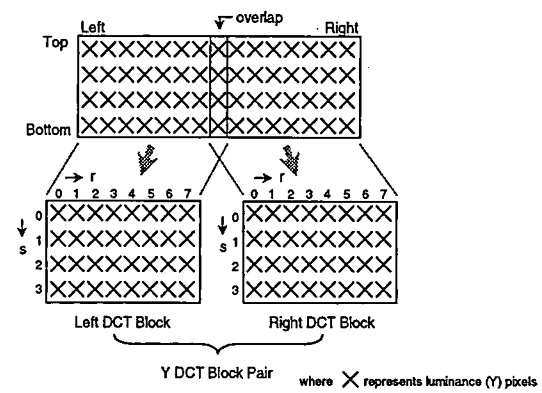

Figure 4 – Overlapped blocking of luminance (Y) pixels

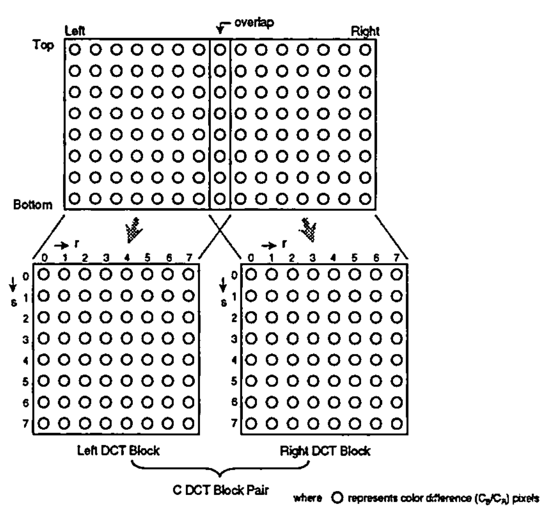

Figure 5 – Overlapped blocking of color-difference C_B/C_R pixels

Let r be the horizontal pixel position number in Y/C DCT block:

r = 0, 1, 2,..., 7.

Let s be the vertical pixel position number in Y/C DCT block:

For Y block, s = 0, 1, 2, 3
For C block, s = 0, 1, 2, ..., 7.

Let P(r,s) be the value of the pixel at the position (r, s).

4.3.1.2 Macro block (MB)

Each macro block (MB) in the 1080i system and the 720p system consists of two Y DCT block pairs, one C_B DCT block and one C_R DCT block. Two Y DCT block pairs are vertically adjacent. The C_B DCT block and C_R DCT block spatially correspond to the two Y DCT block pairs. Four Y DCT blocks (Ya, Yb, Yc, Yd), one C_B DCT block, and one C_R DCT block are shown in figure 6.

4.3.1.3 Super macro block (SMB)

As shown in figure 7, each super macro block (SMB) in the 1080i system and the 720p system consists of two macro blocks which are horizontally adjacent. Two C DCT blocks of C_B/C_R in one super macro block are one C DCT block pair of C_B/C_R.

Let YS be the Y DCT block number in each SMB as shown in figure 7:

YS = 0, 1, 2,..., 7.

Let CS be the C DCT block number in each SMB as shown in figure 7:

CS = 0, 1.

4.3.2 Super macro block arrangement

4.3.2.1 1080i system

The vertical field dimension, 540 pixels long, is not divisible into an integer by the vertical dimension of the SMB, 8 pixels long.

In order to place all SMBs within the 1920 × 540 pixel matrix of the 1080i field, removal and attachment of half-height SMBs are required as shown in figure 8.

1) Y pixels

– Pixels in four rectangular areas of the horizontal pixel position number from (480 × N) to (59 + 480 × N) and from (360 + 480 × N) to (419 + 480 × N) in the active area and line position number from 536 to 539 in the active area shall be moved horizontally 1020 pixel positions to the right, vertically 4 lines to the bottom (N = 0, 1).

– Pixels in two rectangular areas of the horizontal pixel position number from (240 + 480 × N) to (359 + 480 × N) in the active area and line position number from 536 to 539 in the active area shall be moved horizontally 840 pixel positions to the right, vertically 4 lines to the bottom (N = 0, 1).

– Pixels in four rectangular areas of the horizontal pixel position number from (960 + 480 × N) to (1019 + 480 × N) and from (1320 + 480 × N) to (1379 + 480 × N) in the active area and line position number from 536 to 539 in the active area shall be moved horizontally 900 pixel positions to the left, vertically 4 lines to the bottom (N = 0, 1).

– Pixels in two rectangular areas of the horizontal pixel position number from (1 200 + 480 × N) to (1 319 + 480 × N) in the active area and line position number from 536 to 539 in the active area shall be moved horizontally 1080 pixel positions to the left, vertically 4 lines to the bottom (N = 0, 1).

2) C_B/C_R pixels

C_B/C_R pixels occupy the positions held by the even-numbered Y horizontal pixel numbers. The half height SMB replacement operation, identical to the Y pixels as described above, is performed for C_B/C_R pixels.

The arrangement of the SMBs in one field is shown in figure 9. The same horizontal arrangement of 64 SMBs is repeated with 67 SMBs in the vertical direction from the top, and there are 32 SMBs in the horizontal direction at the bottom. The number of SMBs in one field is 4320 as described below:

(vertical 67 SMBs × horizontal 64 SMBs) + 32 SMBs = 4320 SMBs

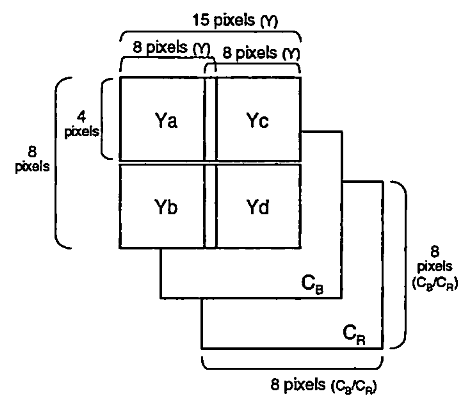

Figure 6 – Macro block structure in 1080i and 720p systems

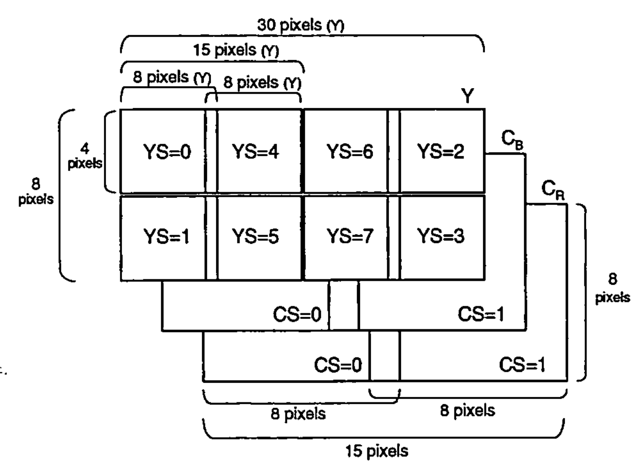

Figure 7 – Super macro block structure in 1080i and 720p systems

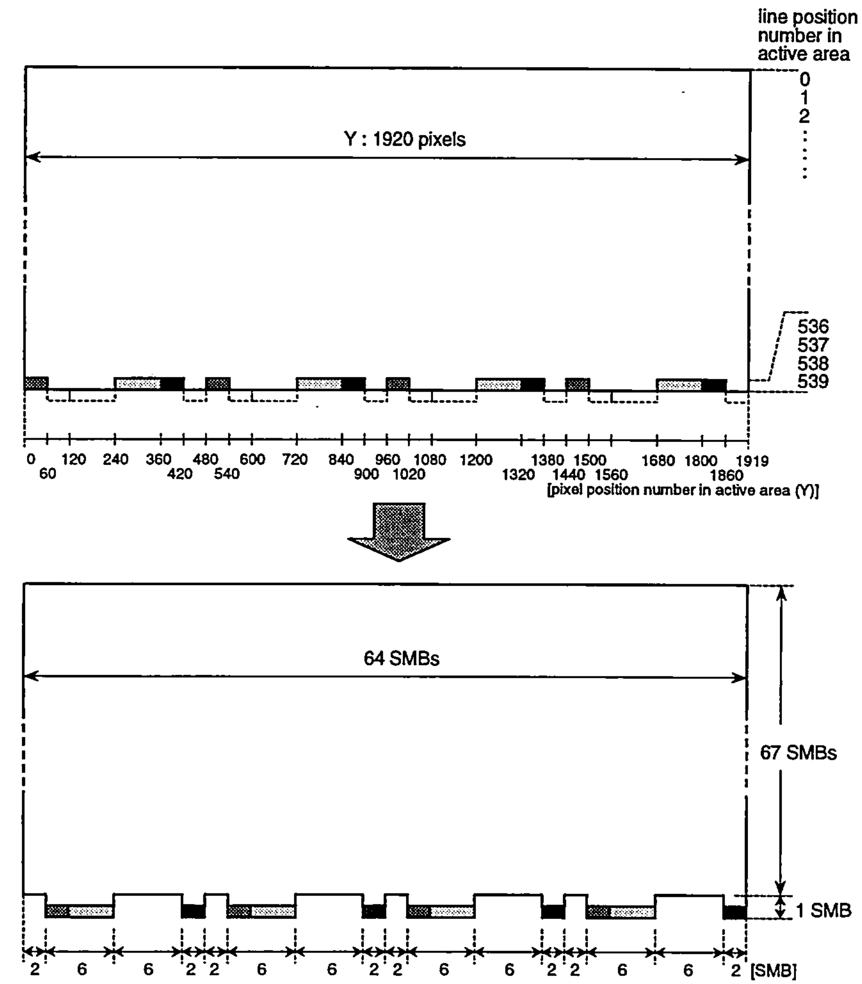

Figure 8 – Pixel arrangement for SMBs of 1080i system

Page 7 of 32 pages Page 8 of 32 pages

4.3.2.2 720p system

As the first step of block formation, 160 dummy of Y pixels and 80 dummy of C_B/C_R pixels are added as rightmost pixels in each line. The value of dummy shall be 040_h for Y and 200_h for C.

The arrangement of SMBs in one frame for the 720p system is shown in figure 10. The same horizontal arrangement of 48 SMBs is repeated with 90 SMBs in the vertical direction from top to bottom. The number of SMBs in one frame is 4320 as described below:

vertical 90 SMBs × horizontal 48 SMBs = 4320 SMBs

4.4 SMBG distribution

4.4.1 1080i system

4320 SMBs in one field are divided into four SMBGs as shown in figure 11.

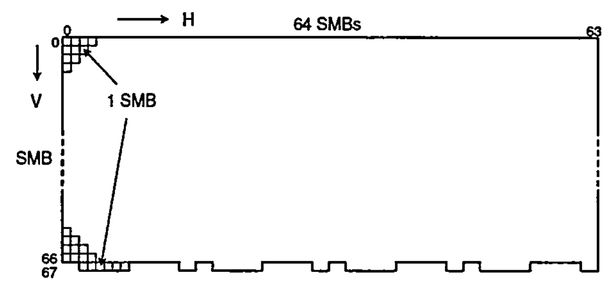

Figure 9 – Arrangement of SMBs in one field for 1080i system

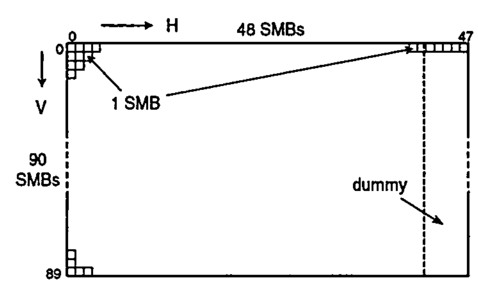

Figure 10 – Arrangement of SMBs in one frame for 720p system

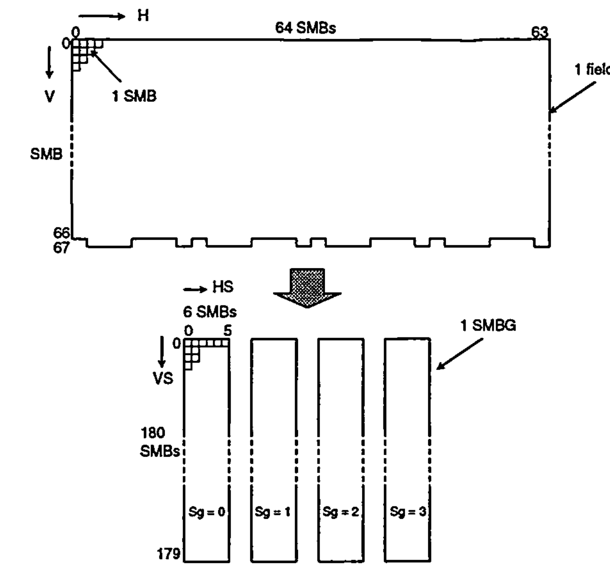

Figure 11 – SMBG distribution in 1080i system

Let H be the horizontal position number of SMB within video field
H = 0, 1, 2,..., 63.

Let V be the vertical position number of SMB within video field
V = 0, 1, 2,..., 67.

Let HS be the column position number of SMB within one SMBG
HS = 0, 1, 2,..., 5.

Let VS be the row position number of SMB within one SMBG
VS = 0, 1, 2,..., 179.

Let Sg be the SMBG number in one field
Sg = 0, 1, 2, 3.

The distribution method is described as follows:

V = (int (VS / 8)) × 3 + int (((VS mod 8) × 6 + HS) / 16)

H = (((f(h, v) + ((int (VS / 8)) - Sg) × 8) mod 32) × 2 + ((int (VS / 8)) mod 2) xnor (HS mod 2)

where v = int (((VS mod 8) × 6 + HS) / 16)
h = int ((((VS mod 8) × 6 + HS) mod 16) / 2)
xnor: exclusive nor

Value of f (h, v)

v \ h	0	1	2	3	4	5	6	7
0	1	2	0	19	20	21	15	14
1	15	10	9	11	30	29	28	24
2	24	25	5	7	6	20	19	18

4.4.2 720p system

4320 SMBs in one frame are divided into four SMBs as shown in figure 12.

Let H be the horizontal position number of SMB within video frame
H = 0, 1, 2,..., 47.

Let V be the vertical position number of SMB within video frame
V = 0, 1, 2,..., 89.

Let HS be the column position number of SMB within one SMBG
HS = 0, 1, 2,..., 5.

Let VS be the row position number of SMB within one SMBG
VS = 0, 1, 2,..., 179.

Let Sg be the SMBG number in one frame
Sg = 0, 1, 2, 3.

The distribution method is described as follows:

V = int (VS/ 2)
H = (VS mod 2) × 24 + ((Sg +f ((int (VS/2)) mod4)) mod 4) x 6 + ((HS-int (VS/ 2)) mod 6)

where f (0) = 0, f (1) = 1, f (2) = 3, f (3) = 2

4.5 DCT

The maximum excursion of all pixel value is changed to fall between −511 and +511 by the subtraction of 512 from the original sampled value. 8 x 4 pixels P(r, s) of each Y DCT block and 8 x 8 pixels P(r, s) of each C_B/C_R DCT block are transformed into 8 x 4 Y DCT coefficients and 8 x 8 C DCT coefficients respectively.

Let t be the horizontal frequency number in Y/C DCT coefficient block as shown in figure 13
t = 0, 1, 2,..., 7.

Let u be the vertical frequency number in Y/C DCT coefficient block
For Y DCT coefficient block, u = 0, 1, 2, 3
For C DCT coefficient block, u = 0, 1, 2,..., 7.

Let C(t, u) be the value of the DCT coefficient at frequency (t, u).

The coefficient of t = 0 and u = 0 is called a DC coefficient. Other coefficients are called AC coefficients.

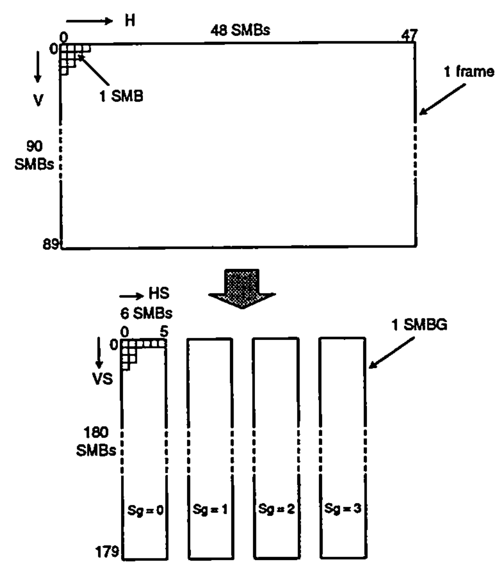

Figure 12 – SMBG distribution in 720p system

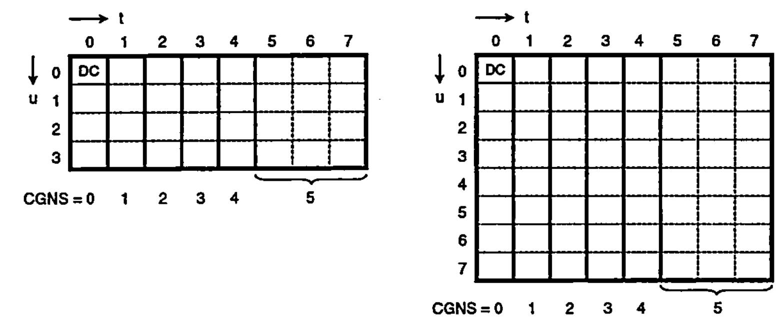

Figure 13 – Structure of DCT coefficient block

4.5.1 DCT/IDCT for Y

DCT/IDCT for the Y signal are defined as shown below:

DCT :

$$C(t,u) = \sqrt{2} \times C1(t) \times C2(u) \sum_{s=0}^{3} \sum_{r=0}^{7} (P(r,s) \times \cos(\pi u (2s+1)/8) \times \cos(\pi t (2r+1)/16))$$

IDCT:

$$P(r,s) = (1/\sqrt{2}) \sum_{u=0}^{3} \sum_{t=0}^{7} (C1(t) \times C2(u) \times C(t,u) \times \cos(\pi u (2s+1)/8) \times \cos(\pi t (2r+1)/16))$$

where $C1(t) = 1/2\sqrt{2}$ for t = 0
$C1(t) = 1/2$ for t = 1 to 7
$C2(u) = 1/2$ for u = 0
$C2(u) = 1/\sqrt{2}$ for u = 1 to 3

The structure of the Y DCT coefficient block is shown in figure 13 a). DCT coefficients C(t, u) of the Y DCT coefficient block are divided into 6 DCT coefficient groups (CGs).

Let CGNS be the CG number as shown in figure 13 a)
CGNS = 0, 1, 2, 3, 4, 5.

4.5.2 DCT/IDCT for C

DCT/IDCT for C (C_B/C_R) signal are defined as shown below:

DCT:

$$C(t,u) = C3(t) \times C4(u) \times C2(u) \sum_{s=0}^{7} \sum_{r=0}^{7} (P(r,s) \times \cos(\pi u (2s+1)/16) \times \cos(\pi t (2r+1)/16))$$

IDCT:

$$P(r,s) = \sum_{u=0}^{7} \sum_{t=0}^{7} (C3(t) \times C4(u) \times C(t,u) \times \cos(\pi u (2s+1)/16) \times \cos(\pi t (2r+1)/16))$$

where $C3(t) = 1/2\sqrt{2}$ for t = 0
$C3(t) = 1/2$ for t = 1 to 7
$C4(u) = 1/2\sqrt{2}$ for u = 0
$C4(u) = 1/2$ for u = 1 to 7

The structure of the C DCT coefficient block is shown in figure 13b). DCT coefficients C(t, u) of the C DCT coefficient block are divided into 6 DCT coefficient groups (CGs).

Let CGNS be the CG number as shown in figure 13b)
CGNS = 0, 1, 2, 3, 4, 5.

4.6 Categorization and weighting

Each MB is categorized into one of the categories and weighting is performed by multiplying all AC coefficients of the subject MB by a weighting function W(t, u) selected by the category. The DC coefficient is not weighted.

There are four categories (CY0, CY1, CY2, CY3) for the Y DCT block, and three categories (CC0, CC1, CC2) for the C_B/C_R DCT block. Each category has its own weighting function. The weighting functions are selectively used to optimize the data compression process by categorization.

4.6.1 Categorization

MB categorization is identified by category flags of FMB, FYa, FYb, FYc, FYd, FC_B, and FC_R. FYa, FYb, FYc, and FYd correspond to the Y DCT blocks of Ya, Yb, Yc, Yd in figure 6, respectively. If the value of the quantized DC coefficient (−255, ..., 0, ..., 255, decimal) of C_B DCT block is less than 24, then FC_B is set to 0, else FC_B is set to 1. If the value of the quantized DC coefficient (−255, ..., 0, ..., 255) of the C_R DCT block is less than 44, then FC_R is set to 0, else FC_R is set to 1.

Categories of CY0, CY1, CY2, and CY3 for the Y signal and categories of CC0 and CC1 for the C signal are expressed by the flags as shown in tables 2 to 4.

Table 2 – Categorization of Y signal

Flag				Category
FMB	FYa, FYb, FYc, FYd	FC_B	FC_R	
1	–	–	–	CY0
0	1	–	–	CY1
0	0	1	–	CY2
0	0	–	1	CY2
0	0	0	0	CY3

where – = arbitrary

Table 3 – Categorization of C_B signal

Flag		Category
FMB	FC_B	
1	–	CC0
0	1	CC1
0	0	CC2

where – = arbitrary

Table 4 – Categorization of C_R signal

Flag		Category
FMB	FC_B	
1	–	CC0
0	1	CC1
0	0	CC2

where – = arbitrary

4.6.2 Weighting

Weighting function W(t,u) in each category is defined below:

4.6.2.1 Y signal

– Category CY0

$W(t, u) = \text{Table CY0}(t,u) \times \cos(0.045\pi t) \times \cos(0.060\pi u) / \sqrt{2}$ (see table 5).

– Category CY1

$W(t, u) = \text{Table CY1}(t,u) \times \cos(0.045\pi t) \times \cos(0.0585\pi u) / \sqrt{2}$ (see table 6).

– Category CY2

$W(t, u) = \text{Table CY2}(t,u) \times \cos(0.045\pi t) \times \cos(0.0585\pi u) / \sqrt{2}$ (see table 7).

– Category CY3

$W(t, u) = \text{Table CY3}(t,u) \times \cos(0.045\pi t) \times \cos(0.0585\pi u) / \sqrt{2}$ (see table 8).

Table 5 – Table CY0(t, u)

u \ t	0	1	2	3	4	5	6	7
0	–	0.25	0.25	0.125	0.125	0.125	0.125	0.125
1	0.25	0.25	0.25	0.125	0.125	0.125	0.0625	0.0625
2	0.25	0.25	0.125	0.125	0.125	0.0625	0.0625	0.0625
3	0.125	0.125	0.125	0.125	0.125	0.0625	0.0625	0.0625

where – = N.A.

Table 6 – Table CY1(t, u)

u \ t	0	1	2	3	4	5	6	7
0	–	0.5	0.5	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$
1	0.5	0.5	0.5	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$
2	0.5	0.5	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$
3	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$

where – = N.A.

Table 7 – Table CY2(t, u)

u \ t	0	1	2	3	4	5	6	7
0	–	1	1	0.5	0.5	0.5	0.5	0.5
1	1	1	0.5	0.5	0.5	0.5	0.5	0.5
2	1	0.5	0.5	0.5	0.5	0.5	0.5	0.5
3	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5

where – = N.A.

Table 8 – Table CY3(t, u)

u \ t	0	1	2	3	4	5	6	7
0	–	0.5	0.5	0.5	0.5	0.5	0.5	0.5
1	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
2	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
3	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5

where – = N.A.

4.6.2.2 C_B/C_R signal

– Category CC0

W(t, u) = Table CC0(t,u) × cos (0.065πt) × cos (0.065πu) (see table 9).

– Category CC1

W(t, u) = Table CC1(t,u) × cos (0.065πt) × cos (0.065πu) (see table 10).

– Category CC2

W(t, u) = Table CC2(t,u) × cos (0.065πt) × cos (0.065πu) (see table 11).

4.7 CG shuffling

In order to improve data robustness against error, weighted DCT CGs are shuffled within the same CGNSs of the 6 SMBs to 12 RMBs as shown in figures 14 and 15. Each RMB comprises 4 Y shuffled DCT coefficient blocks, one shuffled C_B DCT coefficient block, and one shuffled C_R DCT coefficient block.

Table 9 – Table CC0(t, u)

u \ t	0	1	2	3	4	5	6	7
0	–	0.25	0.25	0.125	0.125	0.125	0.125	0.125
1	0.25	0.25	0.125	0.125	0.125	0.125	0.125	0.0625
2	0.25	0.125	0.125	0.125	0.125	0.125	0.0625	0.0625
3	0.125	0.125	0.125	0.125	0.125	0.0625	0.0625	0.0625
4	0.125	0.125	0.125	0.125	0.0625	0.0625	0.0625	0.0625
5	0.125	0.125	0.125	0.0625	0.0625	0.0625	0.0625	0.0625
6	0.125	0.125	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625
7	0.125	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625	0.0625

where – = N.A.

Table 10 – Table CC1(t, u)

u \ t	0	1	2	3	4	5	6	7
0	–	1	1	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$
1	1	1	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$
2	1	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$
3	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$
4	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$
5	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$
6	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$
7	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$	$1/\sqrt{2}$

where – = N.A.

Table 11 – Table CC2(t, u)

u \ t	0	1	2	3	4	5	6	7
0	–	0.5	0.5	0.5	0.5	0.5	0.5	0.5
1	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
2	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
3	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
4	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
6	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
7	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5

where – = N.A.

Let Ycoef (HS, VS, YS, CGNS) be the DCT CG which is referred by CGNS, YS, VS, and HS.

Let HR be the column position number of RMB
HR = 0, 1, 2, ..., 11.

Let VR be the row position number of RMB
VR = 0, 1, 2,..., 179.

Let YR be the Y DCT coefficient block number within one RMB
YR = 0, 1, 2, 3.

Let CGNR be the DCT CG number of one Y DCT coefficient block or one C_B/C_R DCT coefficient block within one RMB
CGNR = 0, 1, 2, ..., 5.

Let YRcoef (HR, VR, YR, CGNR) be the DCT CG which is referred by CGNR, YR, VR, and HR.

The shuffling method of Y DCT CGs is described in the following equations:

For HR = 0 to 5

YRcoef (HR, VR, YR, CGNR) = Ycoef ((CGNR - HR - int (VR / 32)) mod 6, VR, YR + 4 × int (((CGNR - HR) mod 6) / 3), CGNR)

For HR = 6 to 11

YRcoef (HR, VR, YR, CGNR) = Ycoef ((1 - (CGNR + HR + int (VR / 32))) mod 6, VR, YR + 4 × int (((4 - (CGNR + HR)) mod 6) / 3), CGNR)

Let Ccoef(HS, VS, CS, CGNS) be the DCT CG which is referred by CGNS, CS, VS, and HS.

Let CRcoef(HR, VR, CGNR) be the DCT CG which is referred by CGNR, VR, and HR.

The shuffling method of C_B/C_R DCT CGs is described in the following equations:

For HR = 0 to 5

CRcoef(HR, VR, CGNR = Ccoef ((CGNR - HR- int (VR / 32)) mod 6, VR, int (((CGNR - HR) mod 6) / 3), CGNR)

For HR = 6 to 11

CRcoef(HR, VR, CGNR) = Ccoef ((1 - (CGNR + HR + int (VR / 32))) mod 6, VR, int ((4 - (CGNR + HR mod 6) / 3), CGNR)

4.8 RMB shuffling

In order to improve data robustness against error, RMBs are shuffled within the column of 180 RMBs as shown in figure 16.

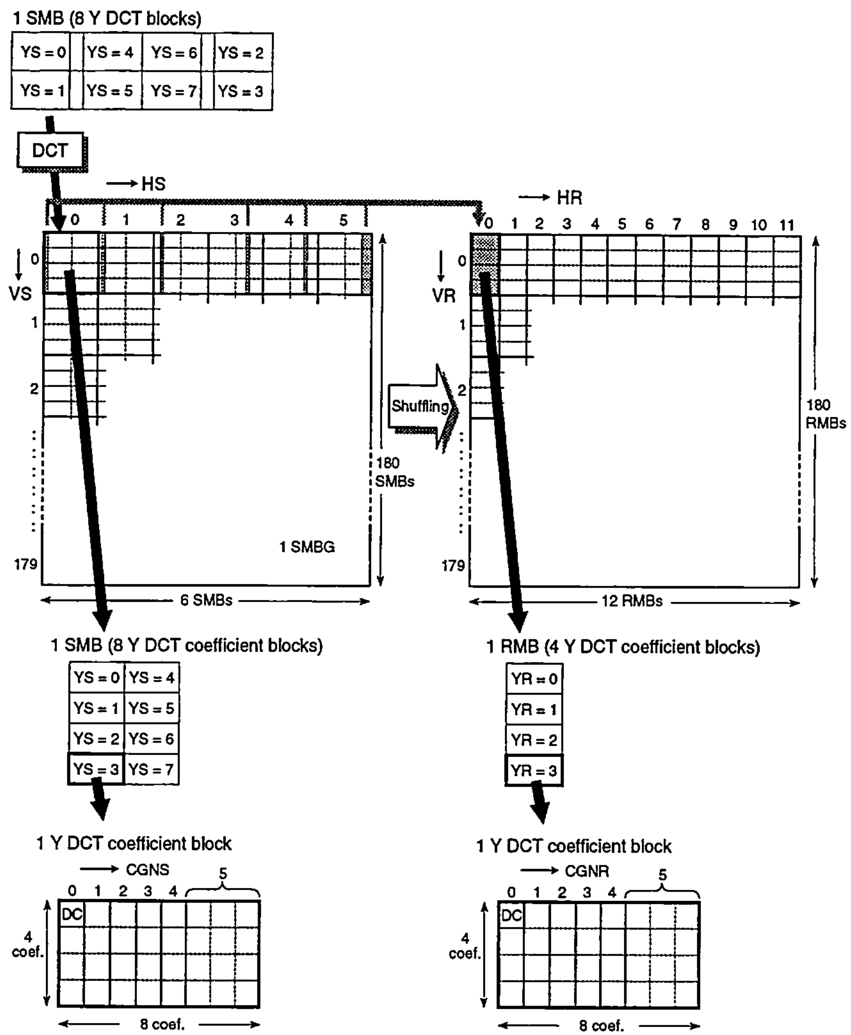

Figure 14 – CG shuffling for Y

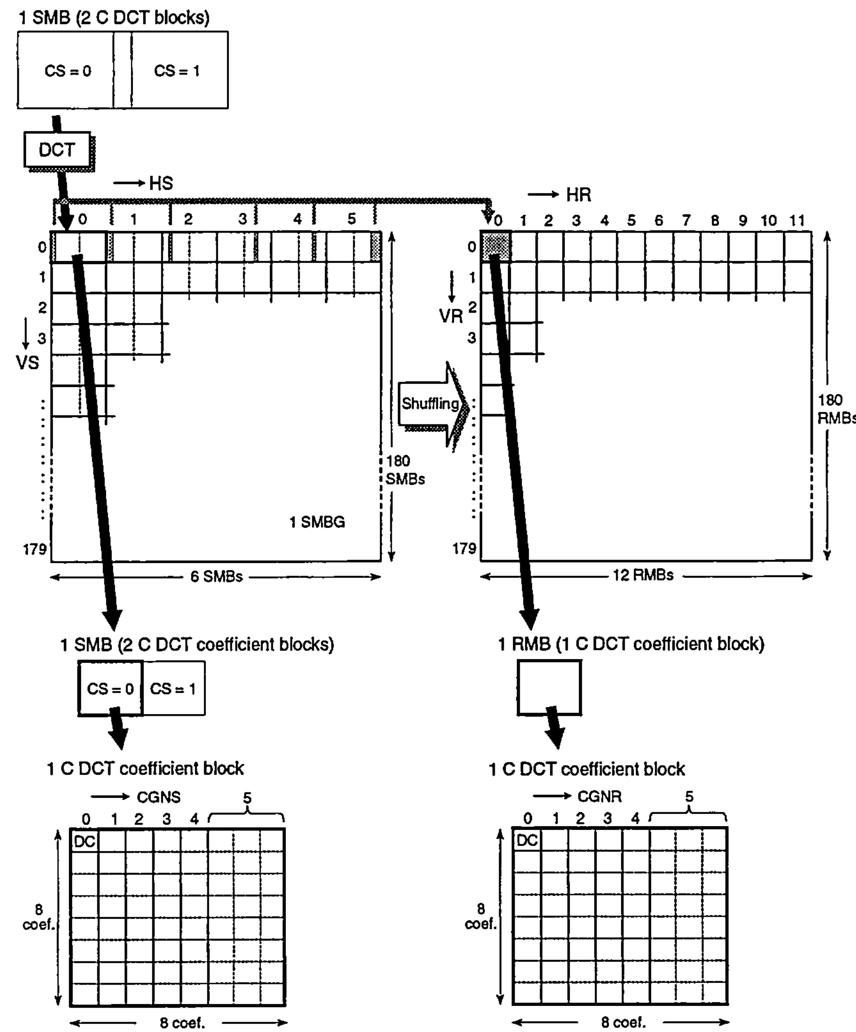

Figure 15 – CG shuffling for C

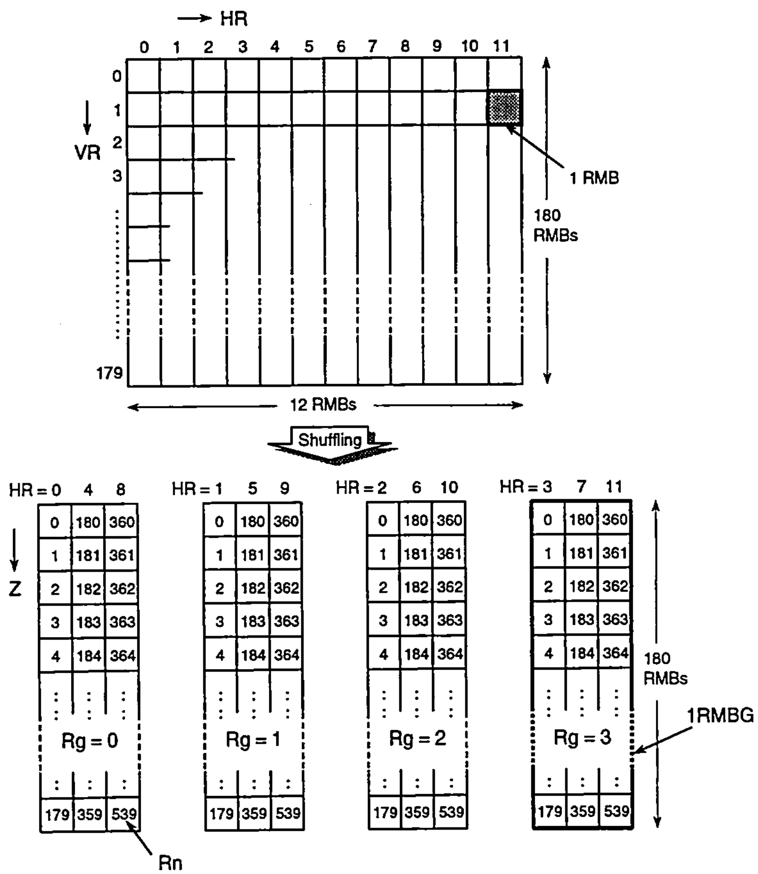

Figure 16 – RMB shuffling

Let Z be the row position number of the RMB after RMB shuffling.

Z = 0, 1, 2, ..., 179.

The method of RMB shuffling is described in the following equation:

$$Z = (17 \times (VR - Offset(HR))) \bmod 180$$

where value of Offset(HR)

HR	Offset (HR)
0	0
1	165
2	150
3	135
4	120
5	105
6	90
7	75
8	60
9	45
10	30
11	15

Then, the RMBs corresponding to one SMBG are divided into 4 RMB groups (RMBGs). There are 540 RMBs in each RMBG.

Let Rg be the RMBG number within the RMBs as shown in figure 16

Rg = 0, 1, 2, 3.

Let Rn be the number of the RMB coding order in each RMBG

Rn = 0, 1, 2,..., 539.

The dividing method is described in the equations:

$$Rg = HR \bmod 4$$
$$Rn = Z + 180 \times int(HR / 4)$$

4.9 Quantization

AC coefficient quantization step values (Q step) are related to the quantization number (Qno) through a relationship as shown below:

$$Qstep = 2^{(Qno \times 6/127 + 1)}$$

where Qno = 0, 1, 2,..., 127

Weighted AC coefficients are divided by Q step to be rounded into a signed value of 12 bits. The Q step for the DC coefficient is always 16. The DC coefficient is rounded into a signed integer value of 9 bits by quantization.

4.10 Rate control

The amount of the compressed data of one RMBG shall be controlled to 30240 bytes or less. Qno is selected every three successive RMBs which are defined in 4.12.1.

4.11 VLC

Variable length coding (VLC) is an operation for transforming quantized AC coefficients in a DCT coefficient block into variable length codes using two-dimensional Huffman coding. One or some successive AC coefficients within a DCT coefficient block are coded into one variable length. The variable length coding process is as follows:

1) AC coefficients in a DCT coefficient block are arranged in order of AC coefficient number i (i = 1 to 31 (Y), 63 (C_B/C_R)) as shown in figure 17.

2) Each nonzero AC coefficient in the arranged AC coefficients is represented by the combination of size and level-codeword as shown in table 12. The j-th nonzero AC coefficient is expressed as follows:

(Size(j), level-codeword(j)), where j = 1, 2, ...

3) Zero-run(j) gives the number of successive AC coefficients quantized to zero between (j-1)th nonzero AC coefficient and j-th nonzero AC coefficient.

The composite value of zero-run(j) / size(j) is Huffman coded to be the run_size-codeword(j) according to table 13.

4) Each run_size-codeword(j) is followed by level-codeword(j) to be one VLC codeword.

Since the number of successive zero coefficients may exceed 15, the codeword of ZRL (111111101100, shown in table 13) is defined to represent 15 successive zero coefficients followed by a coefficient of zero amplitude (zero-run / size = 15 / 0).

→ horizontal frequency number

↓ vertical frequency number

[DC]	4	8	12	16	20	24	28
1	5	9	13	17	21	25	29
2	6	10	14	18	22	26	30
3	7	11	15	19	23	27	31

a) Y DCT coefficient block

→ horizontal frequency number

↓ vertical frequency number

[DC]	8	16	24	32	40	48	56
1	9	17	25	33	41	49	57
2	10	18	26	34	42	50	58
3	11	19	27	35	43	51	59
4	12	20	28	36	44	52	60
5	13	21	29	37	45	53	61
6	14	22	30	38	46	54	62
7	15	23	31	39	47	55	63

b) C DCT coefficient block

Figure 17 – Order of VLC coding

Table 12 – Codewords for variable length coding (1)

Coefficient (signed 12 bits)	Size	Level-codeword (0 bit – 11 bits)
0	0	–
−1, 1	1	0, 1
−3, −2, 2, 3	2	00,01, 10, 11
−7, ..., −4, 4, ..., 7	3	000, ..., 011, 100, ..., 111
−15, ..., −8, 8, ..., 15	4	0000, ..., 0111, 1000, ..., 1111
−31, ..., −16, 16, ..., 31	5	00000, ..., 01111, 10000, ..., 11111
−63, ..., −32, 32, ..., 63	6	000000, ... 111111
−127, ..., −64, 64, ..., 127	7	0000000, ... 1111111
−255, ..., −128, 128, ..., 255	8	00000000, ... 11111111
−511, ..., −256, 256, ..., 511	9	000000000, ... 111111111
−1023, ..., −512, 512, ..., 1023	10	0000000000, ... 1111111111
−2047, ..., −1024, 1024, ..., 2047	11	00000000000, ... 11111111111

Table 13 – Codewords for variable length coding (2)

Zero-run / Size	Code length	Run_size-codeword
0 / 0 (EOB)	4	1010
0 / 1	2	00
0 / 2	2	01
0 / 3	3	100
0 / 4	4	1011
0 / 5	5	11010
0 / 6	7	1111000
0 / 7	8	11111000
0 / 8	10	1111110101
0 / 9	13	111111100010
0 / 10	13	111111100011
0 / 11	17	11111111110111111
1 / 0 (EOM)	17	11111111111111111
1 / 1	4	1100
1 / 2	5	11011
1 / 3	7	1111001
1 / 4	9	111110110
1 / 5	11	11111110011
1 / 6	14	11111111001000
1 / 7	14	11111111001001
1 / 8	14	11111111001010
1 / 9	14	11111111001011
1 / 10	15	111111110011000
1 / 11	17	11111111111000011
2 / 1	5	11100
2 / 2	8	11111001
2 / 3	10	1111110110
2 / 4	12	111111101101
2 / 5	15	111111110011001
2 / 6	15	111111110011010
2 / 7	15	111111110011011
2 / 8	15	111111110011100
2 / 9	15	111111110011101
2 / 10	15	111111110011110
2 / 11	17	11111111111000111
3 / 1	6	111010
3 / 2	9	111110111
3 / 3	12	111111101110
3 / 4	15	111111110011111
3 / 5	15	111111110100000
3 / 6	15	111111110100001
3 / 7	15	111111110100010
3 / 8	15	111111110100011
3 / 9	15	111111110100100
3 / 10	15	111111110100101

Zero-run / Size	Code length	Run_size-codeword
3 / 11	17	11111111111001011
4 / 1	6	111011
4 / 2	10	1111110111
4 / 3	15	111111110100110
4 / 4	15	111111110100111
4 / 5	15	111111110101000
4 / 6	15	111111110101001
4 / 7	15	111111110101010
4 / 8	15	111111110101011
4 / 9	15	111111110101100
4 / 10	15	111111110101101
4 / 11	17	11111111111001111
5 / 1	7	1111010
5 / 2	11	11111110100
5 / 3	15	111111110101110
5 / 4	15	111111110101111
5 / 5	15	111111110110000
5 / 6	15	111111110110001
5 / 7	15	111111110110010
5 / 8	15	111111110110011
5 / 9	15	111111110110100
5 / 10	15	111111110110101
5 / 11	17	11111111111010011
6 / 1	7	1111011
6 / 2	12	111111101111
6 / 3	15	111111110110110
6 / 4	15	111111110110111
6 / 5	15	111111110111000
6 / 6	15	111111110111001
6 / 7	15	111111110111010
6 / 8	15	111111110111011
6 / 9	15	111111110111100
6 / 10	15	111111110111101
6 / 11	17	11111111111010111
7 / 1	8	11111010
7 / 2	13	1111111100000
7 / 3	15	111111110111110
7 / 4	15	111111110111111
7 / 5	15	111111111000000
7 / 6	15	111111111000001
7 / 7	15	111111111000010
7 / 8	15	111111111000011
7 / 9	15	111111111000100
7 / 10	15	111111111000101
7 / 11	17	11111111111011011

(continued)

Table 13 – Codewords for variable length coding (2) (concluded)

Zero-run / Size	Code length	Run_size-codeword
8 / 1	9	111111000
8 / 2	13	1111111100001
8 / 3	15	111111111000110
8 / 4	15	111111111000111
8 / 5	15	111111111001000
8 / 6	15	111111111001001
8 / 7	15	111111111001010
8 / 8	15	111111111001011
8 / 9	15	111111111001100
8 / 10	15	111111111001101
8 / 11	17	11111111111011111
9 / 1	9	111111001
9 / 2	15	111111111001110
9 / 3	15	111111111001111
9 / 4	15	111111111010000
9 / 5	16	1111111110100010
9 / 6	16	1111111110100011
9 / 7	16	1111111110100100
9 / 8	16	1111111110100101
9 / 9	16	1111111110100110
9 / 10	16	1111111110100111
9 / 11	17	11111111111100011
10 / 1	10	1111110100
10 / 2	16	1111111110101000
10 / 3	16	1111111110101001
10 / 4	16	1111111110101010
10 / 5	16	1111111110101011
10 / 6	16	1111111110101100
10 / 7	16	1111111110101101
10 / 8	16	1111111110101110
10 / 9	16	1111111110101111
10 / 10	16	1111111110110000
10 / 11	17	11111111111100111
11 / 1	10	1111111000
11 / 2	16	1111111110110001
11 / 3	16	1111111110110010
11 / 4	16	1111111110110011
11 / 5	16	1111111110110100
11 / 6	16	1111111110110101
11 / 7	16	1111111110110110
11 / 8	16	1111111110110111
11 / 9	16	1111111110111000
11 / 10	16	1111111110111001
11 / 11	17	11111111111101011
2 / 1	11	11111110010
12 / 2	16	1111111110111010
12 / 3	16	1111111110111011
12 / 4	16	1111111110111100
12 / 5	16	1111111110111101
12 / 6	16	1111111110111110
12 / 7	16	1111111110111111
12 / 8	16	1111111111000000
12 / 9	16	1111111111000001
12 / 10	16	1111111111000010
12 / 11	17	11111111111101111
13 / 1	11	11111110101
13 / 2	16	1111111111000011
13 / 3	16	1111111111000100
13 / 4	16	1111111111000101
13 / 5	16	1111111111000110
13 / 6	16	1111111111000111
13 / 7	16	1111111111001000
13 / 8	16	1111111111001001
13 / 9	16	1111111111001010
13 / 10	16	1111111111001011
13 / 11	17	11111111111110011
14 / 1	16	1111111111001100
14 / 2	16	1111111111001101
14 / 3	16	1111111111001110
14 / 4	16	1111111111001111
14 / 5	16	1111111111010000
14 / 6	16	1111111111010001
14 / 7	16	1111111111010010
14 / 8	16	1111111111010011
14 / 9	16	1111111111010100
14 / 10	16	1111111111010101
14 / 11	17	11111111111110111
15 / 0 (ZRL)	12	111111101100
15 / 1	16	1111111111010110
15 / 2	16	1111111111010111
15 / 3	16	1111111111011000
15 / 4	16	1111111111011001
15 / 5	16	1111111111011010
15 / 6	16	1111111111011011
15 / 7	16	1111111111011100
15 / 8	16	1111111111011101
15 / 9	16	1111111111011110
15 / 10	17	11111111110111110
15 / 11	17	11111111111111011

The codeword of EOB (1010, shown in table 13) is used to signal that all remaining AC coefficients in a DCT coefficient block are zero. If the AC coefficient numbered 31(Y), 63 (C_B/C_R) in a DCT coefficient block is not zero, the EOB codeword is bypassed.

EOM (1111111111111111, shown in table 13) codeword is used to terminate the VLC coding of 3 RMBs compulsorily.

4.12 Packing

4.12.1 Compressed data of 3 RMBs

As the first step in packing compressed data into DIF blocks, the compressed data (C3RMB) of every 3 RMBs are made.

Let CN be the C3RMB number in one RMBG.
CN = 0, 1, 2, ..., 179.

Let C3RMB[Sg][Rg][CN] be the C3RMB numbered CN of RMBG numbered Rg in SMBG numbered Sg.

Let C3RMB[Sg][Rg][CN][i] be the i-th byte of data in the C3RMB [Sg][Rg][CN] (i = 0, 1, 2, ...).

Each C3RMB is composed of 3 RMBs which are numbered Rn = 3CN, 3CN + 1, 3CN + 2 in each RMBG. The structure of the C3RMB is shown in figure 18. The C3RMB is composed of a fixed length data portion and a variable length data portion. The fixed length data portion comprises side information and DC data. The variable length data portion comprises AC data.

4.12.1.1 Side Information and DC data

The details of the fixed length data portion in figure 18 are as follows:

- SABM: Upper or lower byte of the starting address (2 bytes) of the remainder data of the C3RMB stored in buffer memory. (See 4.12.3 for details.)

- FFL: Field number flag
  0 = the 1st field, 1 = the 2nd field (1080i system);
  N.A. (720p system)

- Qno: 7-bit data of quantization number

- Category flag:

  FMB – FMB flag of the MB, to which the C_B/C_R DC coefficient of the RMB belongs before CG shuffling.

  FMB′ – FMB flag of the other MB in the SMB, to which the MB mentioned above belongs.

  FYa-d – Flags of the four DCT blocks (Ya-Yd) of the MB, to which the C_B/C_R DC coefficient of the RMB belongs before CG shuffling.

  FYa′-d′ — Flag of each DCT block (Ya-Yd) of the other MB in the SMB, to which the MB mentioned above belongs.

  FCB′ – FCB Flag of the C_B DCT block which overlaps the C_B DCT block, to which the C_B DC coefficient of the RMB belongs before CG shuffling.

  FCR′ – FCR Flag of the C_R DCT block which overlaps the C_R DCT block, to which the C_R DC coefficient of the RMB belongs before CG shuffling.

- DC data: 9-bit DC coefficient data of the 18 DCT coefficient blocks in 3 RMBs; 9-bit DC coefficient data are separated into two groups of 8 MSB data and LSB data, respectively. DC data are composed of one sign bit data (MSB, 0: plus or zero, 1: minus) and 8-bit binary data (absolute value).

NOTE – The side information does not include FC_B and FC_R because they can be made from the DC data of the C_B/C_R DCT block.

4.12.1.2 AC data of 3 RMBS

To store the significant data of C3RMB in one DIF block, VLC code words of all DCT coefficient blocks within every 3 RMBs are rearranged. The rearranging method is described below:

```
while (code_word_rearranging_not_finished_in_3_RMB) {
  for (blk_id=0; blk_id < 6 ; blk_id++) {
    for (rmb=0; rmb < 3; rmb++) {
      if (block[rmb][blk_id]_is_not_empty) {
      get_a_code_word_and_connect (see figure 19) }
    }
  }
}
```

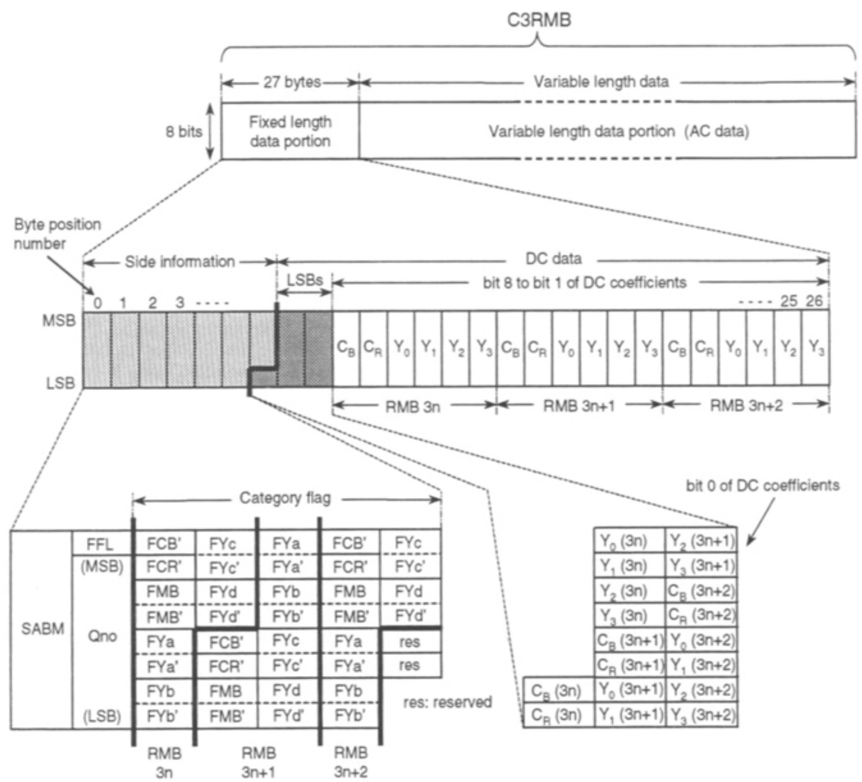

Figure 18 – Structure of C3RMB

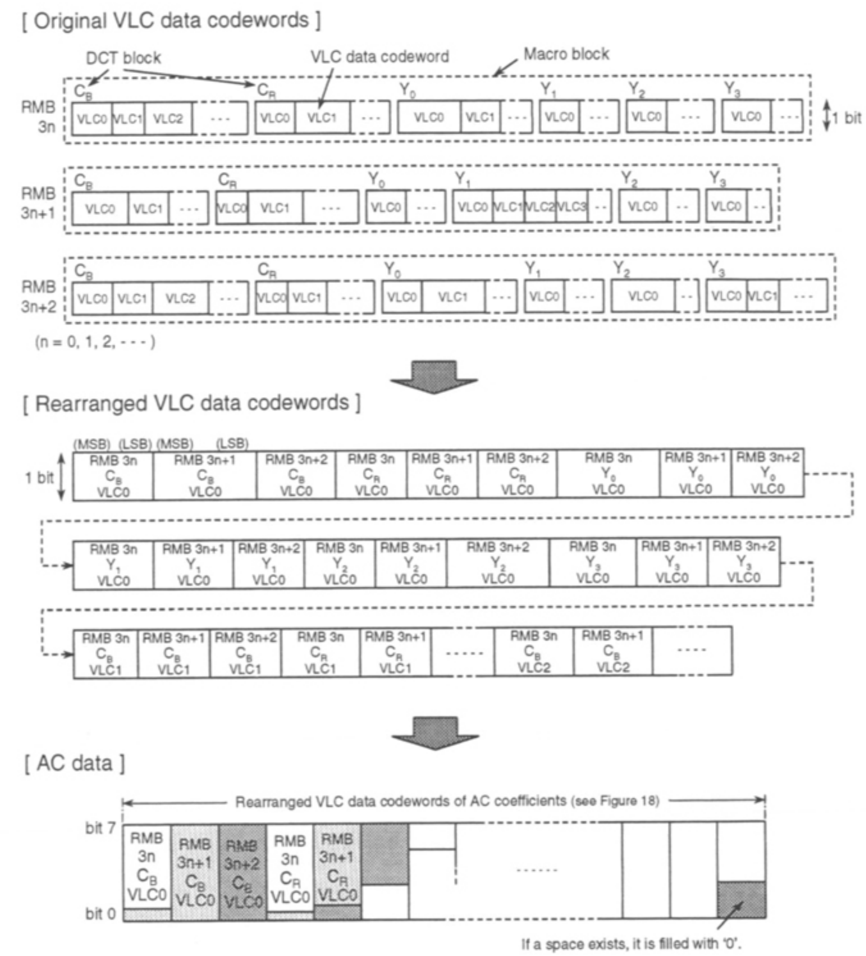

Figure 19 – Rearrangment of VLC data codewords

where blk_id = 0 (C_B - block)
= 1 (C_R - block)
= 2 (Y_0 - block)
= 3 (Y_1 - block)
= 4 (Y_2 - block)
= 5 (Y_3 - block)

The rearranged code words are sequentially connected and divided into bytes, then stored in the AC variable length data portion of the C3RMB. The codewords are arranged from the upper-left part of the variable length data portion according to the order of rearranged VLC codewords as shown in figure 19. After the codewords have been filled into the variable length data portion, there will be less than one byte of empty space remaining. Then the empty space must be filled with bit 0.

The byte length of AC data is int ((the total bit amount of VLC codewords of 3RMBs + 7) / 8); the byte length of compressed 3 RMB data is int ((the total bit amount of VLC codewords of (3RMBs + 7) / 8) + 27; and its maximum value is 768 bytes.

Let LEN[Sg][Rg][CN] be the byte length of C3RMB[Sg][Rg][CN].

4.12.2 DIF block

The 5760 DIF block data stream contains compressed video data for one 1080i field or one 720p frame. The data structure of the HD compressed video stream is shown in figure 20. DIF blocks are numbered 0 to 5759 in order. Each DIF block consists of 85 bytes of data. A 12-byte area in byte position numbers 0 to 11 of the 85 bytes in each DIF block numbered in multiples of 12 (DIF block number DN is 0, 12, ..., 5748) is reserved for data transmission.

The DIF blocks numbered DN = 4×J+2, 4×J+3 (J=0, 1, 2, 3,..., 1439) are called main data DIF blocks and are used to transmit significant data of C3RMB. One main data DIF block has all or most of the data of one C3RMB. The DIF blocks numbered DN = 4×J, 4×J+1 are called remainder data DIF blocks and are used to transmit the remainder of the C3RMB.

Let DIF[M] be the DIF block numbered m where m = 0, 1, 2, 3,..., 5759.

Let DIF[m][i] be the i-th byte of data in DIF[M] where i = 0, 1, 2, ..., 84.

4.12.3 Main data DIF block packing

A pair (packing pair) of compressed data, C3RMB[Sg][Rg][2×K] and C3RMB[Sg][Rg][2×K+1], are packed into a pair of main data DIF blocks, DIF[4×J+2] and DIF[4×J+3] respectively, where K is 0, 1, 2,..., 89 in one RMBG. J is described as follows:

$$J = 360 \times Rg + 4 \times K + (Rg + Sg) \bmod 4$$

If the data of C3RMB overflow their own DIF block, the vacant space in the other DIF block of the pair and buffer memory are used for overflow data storing. The size of the buffer memory for one RMBG is 14940 bytes.

Let BUF[Sg][Rg] be the buffer memory for the RMBG numbered Rg the SMBG numbered Sg.

Let BUF[Sg][Rg][i] be the byte data addressed i in the BUF[Sg][Rg], where i = 0, 1, ..., 14939.

The C3RMB data stored in the main data DIF blocks are called main data and the C3RMB data stored in the buffer memory are called remainder data. After main data DIF block packing, the remainder data of one

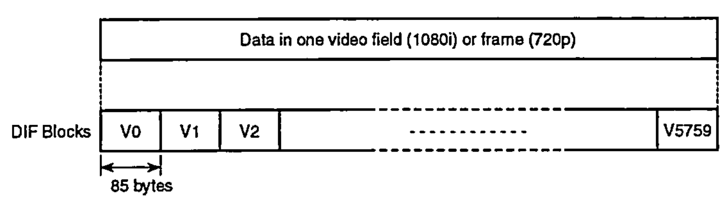

Figure 20 – Data structure of one 1080i field/720p frame

RMBG in buffer memory are packed into 180 remainder data DIF blocks as described in 4.12.4.

Let SA(Sg)[Rg][K) be the starting address of the remainder data of C3RMB[Sg][Rg][2×K] and/or C3RMB[Sg][Rg][2×K+1] in the buffer memory.

So that the remainder data of C3RMB[Sg][Rg][2×K] and/or C3RMB[Sg)[Rg)[2×K+1] are stored at the addresses from SA[Sg][Rg][K] to (SA[Sg][Rg][K+1]–1) in the buffer memory, SA[Sg][Rg][0] is always 0000_h, and SA[Sg][Rg)[90] is the next address of the last remainder data of each RMBG in the buffer memory.

Before the main data packing in DIF[4×J+2] and DIF[4×J+3], the most significant byte and the least significant byte of the data SA[Sg][Rg][K](2 byte) are stored in C3RMB[Sg][Rg][2×K][0] and C3RMB[Sg][Rg][2×K+1][0] respectively as the SABM described in 4.12.1.

Let Set_SABM_of_C3RMB(Sg,Rg,K,SA[Sg][Rg][K] be the SABM setting process mentioned above.

As an exception, the address data SA[Sg][Rg][90] are stored in C3RMB[Sg][Rg)[0][0] and C3RMB[Sg][Rg][1][0], in other words, in DIF[4×J+2][0] and DIF[4×J+3][0], where K is 0, as SABM instead of SA[Sg][Rg][0].

Let Set_SA90(Sg,Rg,K,4×J) be the process of SA[Sg][Rg][90] storing mentioned above.

There are four cases in the packing of C3RMB[Sg][Rg][2×K] and C3RMB[Sg][Rg][2×K+1] according to their byte length. The packing method of four cases is described below:

– Case A

In the case that (LEN[Sg][Rg][2×K] = < 85) and (LEN[Sg][Rg][2×K+1] = < 85), all data of C3RMB[Sg][Rg][2×K] and C3RMB[Sg][Rg][2×K+1] are packed into DIF[4×J+2] and DIF[4×J+3] respectively. In this case, the vacant space is not used. The packing method of case A is described below and shown in figure 21.

```
for (i=0; i < LEN [Sg][Rg][2×K]; i++)
   DIF[4×J+2][i] = C3RMB[Sg][Rg][2×K][i];

for (i=0; i < LEN [Sg][Rg][2×K+1]; i++)
   DIF[4×J+3][i] = C3RMB[Sg][Rg][2×K+1][i];

SA[Sg][Rg][K+1]=SA[Sg][Rg][K];
```

– Case B

In the case that ((LEN[Sg][Rg][2×K] = > 85) and (LEN[Sg][Rg][2×K+1] > 85)) or ((LEN[Sg][Rg][2×K] > 85) and (LEN[Sg][Rg)[2×K+1] = > 85), the overflow data of C3RMB[Sg][Rg][2×K] and C3RMB[Sg][Rg][2×K+1] are stored in buffer memory, as shown in figure 21. The packing method of case B is described below:

```
for(i=0; i < 85; i++)
   DIF[4×J+2][i] = C3RMB[Sg][Rg][2×K][i];
ra = SA[Sg][Rg][K];
for(i=0; i < LEN[Sg][Rg][2×K]–85); i++)
   BUF[Sg][Rg][ra+i] = C3RMB[Sg][Rg][2×K][85+i];

ra = ra+(LEN[Sg][Rg][2×K]–85);
for(i=0; i < 85; i++)
   DIF[4×J+3][i] = C3RMB[Sg][Rg][2×K+1][i];
for(i=0; i < (LEN[Sg][Rg][2×K+1]–85); i++)
   BUF[Sg][Rg][ra+i] = C3RMB[Sg][Rg][2×K+1][85+i];

SA[Sg][Rg][K+1]=SA[Sg][Rg][K]+LEN[Sg][Rg][2×K]
+LEN[Sg][Rg][2×K+1]–170.
```

– Case C

In the case that (LEN[Sg][Rg)[2×K] < 85) and (LEN[Sg][Rg][2×K+1] > 85), the overflow data of C3RMB[Sg][Rg][2×K+1] are packed into the vacant space of DIF[4×J+2], and in the case of (LEN[Sg][Rg][2×K+1] – 85) > (85-LEN[Sg][Rg][2×K]), stored in buffer memory as shown in figure 21. The packing method of case C is described below:

```
for(i=0; i < LEN [Sg][Rg][2×K]; i++)
   DIF[4×J+2][i] = C3RMB[Sg][Rg][2×K][i];

for(i=0; i < 85; i++)
   DIF[4×J+3][i] = C3RMB[Sg][Rg][i];
if ((LEN[Sg][Rg][2×K+1] – 85) = < (85-LEN[Sg][Rg][2×K])) {
   for(i=0; i < (LEN[Sg][Rg][2×K+1]–85); i++)
      DIF[4×J+2][LEN[Sg][Rg][2×K]+i] =
                  C3RMB[Sg][Rg][85+i];
   SA[Sg][Rg][K+1]=SA[Sg][Rg][K];
} else {
   for (i=0; i<(85-LEN[Sg][Rg][2×K]); i++)
      DIF[4×J+2][LEN[Sg][Rg][2×K]+i] =
                  C3RMB[Sg][Rg][85+i];
   ra=SA[Sg][Rg][K];
   for(i=0; i<(LEN[Sg][Rg][2×K]+LEN[Sg][Rg][2×K+1]–170); i++)
      BUF[Sg][Rg][ra+i] = C3RMB[Sg][Rg][2×K+1]
        [170-LEN[Sg][Rg][2×K]+i];
```

```
SA[Sg][Rg][K+1] = SA[Sg][Rg][K]+ LEN[Sg][Rg][2×K]
    +LEN[Sg][Rg][2×K+1]-170;
}
```

– Case D

In the case that (LEN[Sg][Rg][2×K] > 85) and (LEN[Sg][Rg][2×K+1] < 85), the overflow data of C3RMB[Sg][Rg][2×K] are packed into the vacant space of DIF[4×J+3], and in the case of (LEN[Sg][Rg][2×K] – 85) > (85-LEN[Sg][Rg][2×K+1]), stored in buffer memory as shown in figure 21. The packing method of case D is described below:

```
for(i=0; i<LEN[Sg][Rg][2×K+1]; i++)
  DIF[4×J+3][i] = C3RMB[Sg][Rg][2×K+1][i];

for(i=0; i<85; i++)
  DIF[4×J+2][i] = C3RMB[Sg][Rg][2×K][i]
if((LEN[Sg][Rg][2×K]-85) = < (85-LEN[Sg][Rg][2×K+1]){
  for(i=0; i < (LEN[Sg][Rg][2×K]-85); i++)
    DIF[4×J+3][84-i] = C3RMB[Sg][Rg][2×K][85+i];
  SA[Sg][Rg][K+1]=SA[Sg][Rg][K];
} else {
  for(i=0; i<(85-LEN[Sg][Rg][2×K+1]); i++)
    DIF[4×J+3][84-i] =
      C3RMB[Sg][Rg][2×K][LEN[Sg][Rg][2×K]+LEN
      [Sg][Rg][2×K+1]-85+i];
  ra=SA[Sg][Rg][K];
  for(i=0; i<(LEN[Sg][Rg][2×K]+LEN[Sg][Rg][2×K+1]
  -170); i++)
    BUF[Sg][Rg][ra+i] = C3RMB[Sg][Rg][2×K][85+i];
  SA[Sg][Rg][K+1]=SA[Sg][Rg][K]+LEN[Sg][Rg][2×K]+
  LEN[Sg][Rg][2×K+1]-170;
}
```

Let PACK_C3RMB_PAIR-IN-DIF(Sg,Rg,K,4×J) be the process of the four cases mentioned above.

The method of packing 90×16 pairs of C3RMBs of one 1080i field or one 720p frame into 90×16 pairs of main data DIF blocks is described below:

```
for (Rg=0; Rg<4; Rg++){
  for (Sg=0; Sg<4; Sg++){
    SA[Sg][Rg][0] = 0;
    for (K=0; K<90; K++){
      J = 360 × Rg + 4 × K + (Rg+Sg) mod4;
      Set_SABM_of_C3RMB(Sg,Rg,K,SA[Sg][Rg][K]);
      PACK_C3RMB_PAIR_IN_DIF (Sg,Rg,K,4×J);
      if (K==89){
        K = 0;

        J = 360 × Rg + 4 x K + (Rg+Sg) mod4;
        Set_SA90 (Sg,Rg,K,4xJ);
      }
    }
  }
}
```

4.12.4 Remainder data DIF block packing

The method of packing the remainder data into one remainder data DIF block is described below:

```
if (((4×J)mod12)==0) { SIZE=73; SKIP=12;
} else {                SIZE=86, SKIP=0;
}
for (i; i<SIZE; i++) DIF[4×J][SKIP+i] = BUF[Sg][Rg][ra+i];
ra=ra+SIZE;
```

Let PACK_REMAINDER_IN_DIF (Sg,Rg,4×J,ra) be the process mentioned above.

All remainder data in the buffer memory of one 1080i field or one 720p frame are packed into 180×16 remainder data DIF blocks. The method of packing remainder data into remainder data DIF blocks is described below:

```
for (Rg=0; Rg<4; Rg++){
  for (Sg=0; Sg<4; Sg++){
    ra = 0;
    for (K=0; K<90; K++){
      J=360 × Rg + 4 × K + (Rg+Sg) mod4;
      PACK_REMAINDER_IN_DIF (Sg,Rg,4×J, ra);
      PACK_REMAINDER_IN_DIF (Sg,Rg,4×J+1, ra);
    }
  }
}
```

The result of packing all the compressed data C3RMBs of one 1080i field or one 720p frame into 5760 DIF blocks (2880 main data DIF blocks and 2880 remainder data DIF blocks) is shown in figure 22.

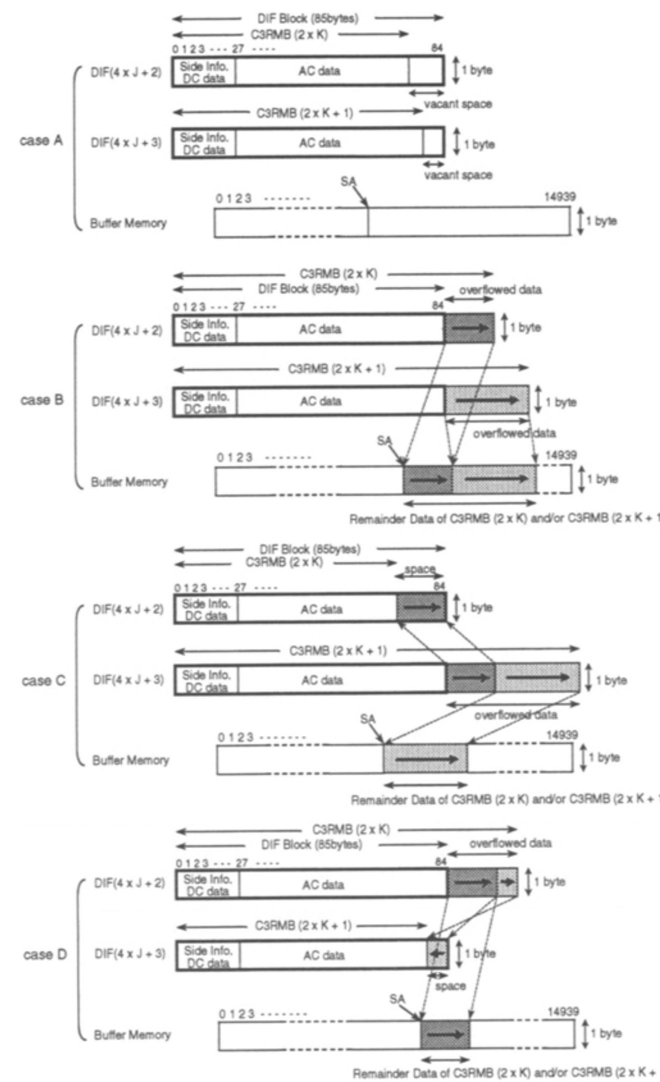

Figure 21 – Main data DIF block packing

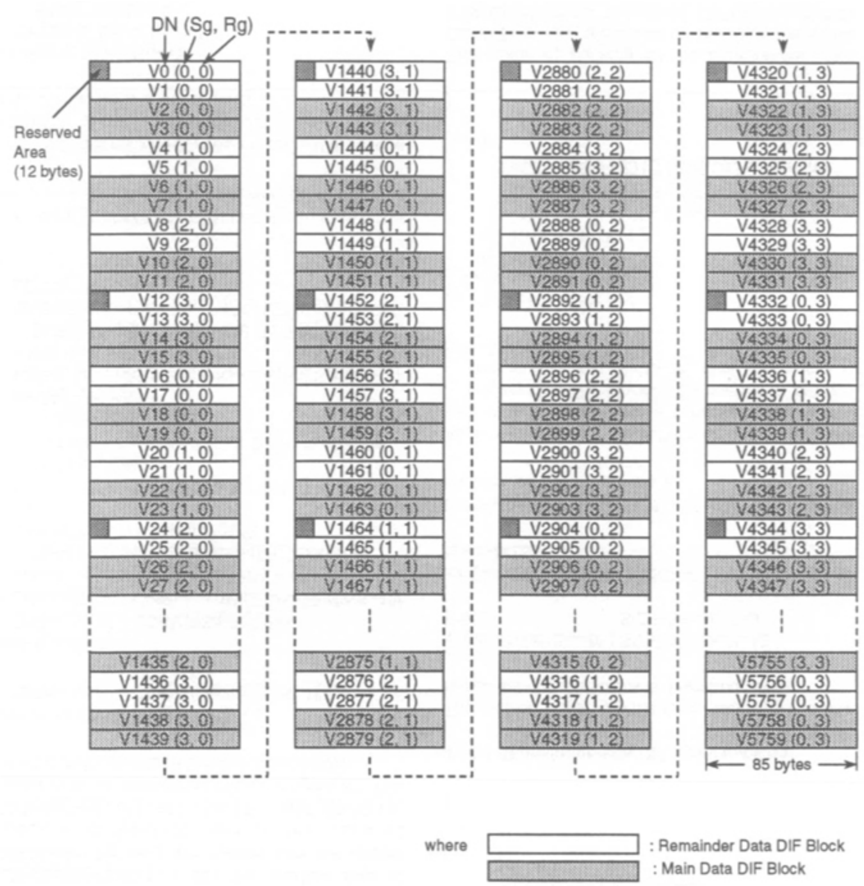

Figure 22 – Packing compressed data into 5760 DIF blocks

Annex A (informative)
Overlapped block DCT coding for robustness

The process of reproduction of missing (error-detected) coefficient in overlapped block DCT coding is shown in figure A.1.

A.1 Missing one column of data in a DCT coefficient block

IDCT: $[P_{m,n}] = [D_{m,m}]^T [A_{m,n}] [D_{n,n}]$, $[Q_{m,n}] = [D_{m,m}]^T [B_{m,n}] [D_{n,n}]$ where [D]: DCT basis matrix.

Pixel data in the most right-side column of the DCT block P: $[P_{m,7}] = [D_{m,m}]^T [A_{m,n}] [D_{n,7}]$

Pixel data in the most left-side column of the DCT block Q: $[B_{m,0}] = [D_{m,m}]^T [B_{m,n}] [D_{n,0}]$

By overlapped blocking: $[P_{m,7}] = [Q_{m,0}]$

From the equations above mentioned: $[A_{m,n}] [D_{n,7}] = [B_{m,n}] [D_{n,0}]$

The relationship of DCT coefficient between [Am,n] and $[B^{err}_{m,n}]$: $[Am,n] [Dn,7] = [B^{err}_{m,n}] [D_{n,0}] + [B_{m,i}] [D_{i,0}]$

where i = 0 to 7, $[B^{err}_{m,n}]$: $[B_{m,n}]$ in which coefficients of column $[B_{m,i}]$ are all replaced to value 0.

Reproduced missing coefficient data $[B_{m,i}]$ (column i in the $[B_{m,n}]$): $[B_{m,i}] = ([A_{m,n}] [D_{n,7}] - [B^{err}_{m,n}] [D_{n,0}]) / [D_{i,0}]$

Reproduced DCT coefficient $[B_{m,n}]$: $[B^{err}_{m,n}] = [B^{err}_{m,n}] + [B_{m,i}]$

A.2 Missing more than one column of data in a DCT coefficient block

All data in error-detected columns are replaced with the value zero. Then the only lowest frequency column in the error-detected columns is reproduced by the same equations described above, supposing all other columns replaced with value zero are originally zero.

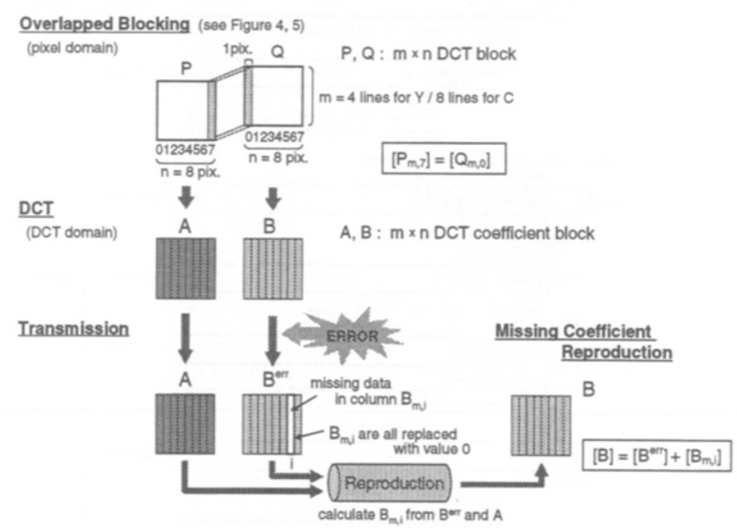

Figure A.1 – Process of missing coefficient reproduction