Cell and CellB Bytestream Definitions

D

This appendix provides an overview of the Cell technology, then focuses on the descriptions of the codes that are used in a Cell bytestream. Although we include some rationale for the existence of the various codes, we do not provide specific implementation information about Cell encoders or decoders.

Introduction to Cell

The Cell image compression technology, which was developed by Sun Microsystems, provides high quality, low bit-rate image compression at low computational cost. Applications where Cell compression can be used include videoconferencing, media distributions on CD-ROM, and multimedia mail applications. The bytestream that is produced by a Cell encoder is variable length and is made up of instructional codes and information about the compressed video image.

There are two versions of the Cell image compression technology: Cell and CellB. Cell compression, which is designed for use with movies, is computationally asymmetric; it takes longer to compress video data than it does to decompress the data. To provide high-quality images, Cell supports Adaptive Colormap Selection, which enables an encoder to change colormaps dynamically. The bytestream of images that are compressed with Cell can be decompressed in software on any SPARCstation™ workstation. Depending on compression ratios, 640x480-resolution movies can be played back at 30 frames / second (fps). Multiple smaller movies can be displayed simultaneously at this rate.

D

CellB, which is derived from Cell, is designed for use in videoconferencing applications. To reduce computational overhead and meet the timing demands of videoconferencing, CellB compression is more computationally symmetric than Cell. CellB uses a fixed colormap and is designed to use vector quantization techniques. The bytestream of a CellB image is simpler (and more compact) than that of a Cell image, which reduces requirements on network bandwidth.

Encoding Images for Cell

Cell works with 3-band RGB images, with no subsampling, and requires that the width and height of the images be divisible by four. CellB takes industry-standard 3-band YC_bC_r video as input.

A cell encoder breaks the video into cells. A cell is 16 pixels, arranged in a 4-by-4 group (see Figure D-1). Cells are encoded into the bytestream in scanline order, from left to right and from top to bottom.

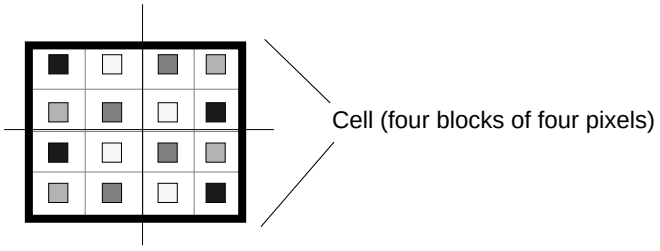


Figure D-1 Cell

The basic encoding scheme used in both versions of Cell is based on an image coding method called Block Truncation Coding (BTC). The 16 pixels in a cell are represented by a 16-bit mask and two colors. The values in the mask specify which color to place at each of the pixel positions. The mask and colors can be chosen to maintain certain statistics of the cell or to reduce contouring in a manner similar to ordered dither.

The primary advantage of BTC is that its decoding process is similar to the operation of character fonting in a color frame buffer. The character display process for a frame buffer takes as input a foreground color, a background color, and a mask that specifies whether to use the foreground or background

color at each pixel. Because this function is so important to the window system, it is often implemented as a display primitive in graphics accelerators. The Cell compression technique leverages these existing primitives to provide full-motion video decoding without special hardware or modifications to the window system.

The basic component of the Cell and CellB bytestreams is the four-byte cell code. The first two bytes of the cell code are a bitmask (Figure D-2). Each bit in the mask represents a pixel in a 16-bit cell. The bitmask is normalized so that the most significant bit is 0. The figure shows the relationship of the bits in the mask to the location of the pixels in a cell. A value of 0 in a mask bit means that the pixel is rendered in the background color (color 0). A value of 1 means that the pixel is rendered in the foreground color (color 1).

The last two bytes of a cell code establish a pixel's color. Cell and CellB differ in the way that pixel colors are derived. In Cell, the values in the color 0 and color 1 bytes are indexes into a colormap. In CellB, the values are indexes into Y/Y and C_b/C_r vector quantization tables.

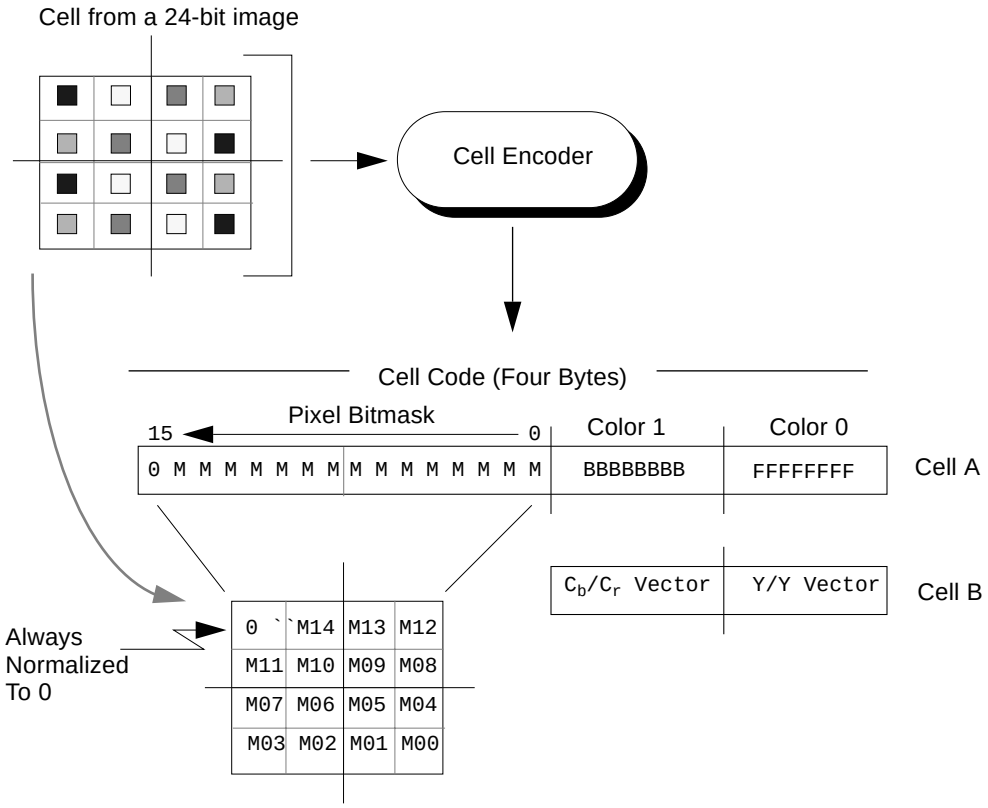


Figure D-2 Encoding a Cell

Cell Bytestream Description

The Cell bytestream can contain the following codes:

- Key-frame headers
- Key parameters
- Cell codes
- Run length codes
- Escape codes

At a minimum, the Cell bytestream must contain a key-frame header, a key frame, and cell codes. More compression is provided by using run length codes for cells. The escape codes enable you to skip frames, load colormaps, change masks and colors, and include non-video (user) data in the bytestream.

The codes are described in the following sections.

Key-Frame Header and Key Parameters

A Cell bytestream contains periodic “key” frames, which are free of interframe escape codes. Every key frame is preceded by a key frame header, which is described below. Immediately after the key frame header, the colormap is encoded. If the colormap has not changed since the previous frame, it is repeated anyway. The first frame of a Cell movie is always a key frame.

Key-Frame Header

The key frame header begins with a series of 8 bytes of all 1’s (0xff). This code is followed by a number of parameters, which are introduced by 1-byte codes. The end of a key frame header is marked by an all-zero byte.

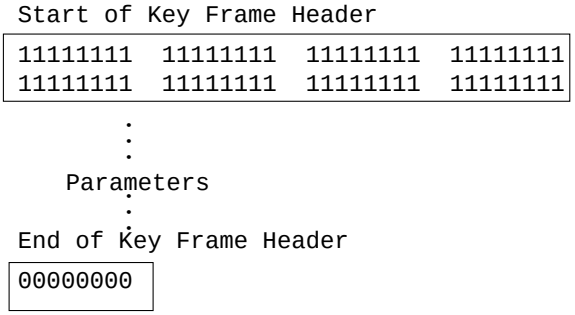
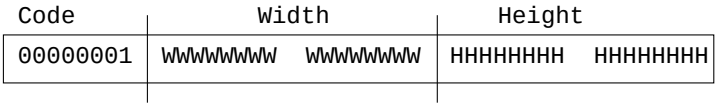


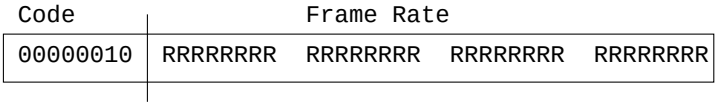
Image Dimensions Key-Frame Code

The following code precedes the 16-bit unsigned width and height of an image.



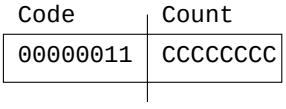
Frame Rate Key-Frame Code

The following code precedes a 32-bit unsigned number that sets the frame rate in microseconds per frame.



Size of Colormap Key-Frame Code

The following code precedes a byte that specifies the maximum size (Count+1) of a colormap in the bytestream.



Reserved Key-Frame Codes

The key frame header codes in the range 00000000 to 11111111 (inclusive) are reserved.

Example Key-Frame Header

As an example, a key frame header for a 320-by-240 movie with a frame rate of 30 fps and a maximum colormap size of 240 would look like this (in hex):

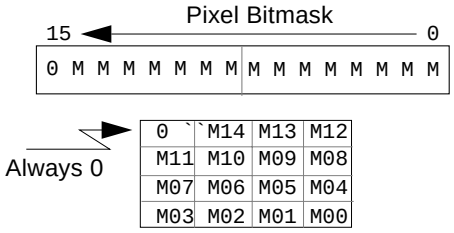
```
ff ff ff ff ff ff ff ff      Start of key frame header
01 01 40 00 f0               Dimensions 320 (0x140) by 240 (0xf0)
02 00 00 82 35               33,333 (0x8235) usecs per frame
03 ef                        Max colormap size is 239 (0xef) + 1
00                           End of key frame header
c0 ...                       Escape code for colormap
```

Cell Code

The four-byte code that describes a 4-by-4 Cell cell is shown below. The values in the Color1 and Color0 bytes are indexes into the current colormap. If a pixel's bit is set, the color that is indexed in the Color1 byte is used. If not set (0), the pixel is rendered with the color that is indexed by the value of the Color0 byte.

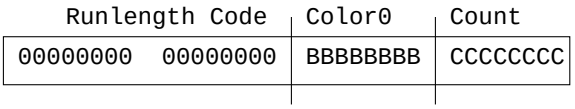
4x4 Bitmask	Color1	Color0
0MMMMMMM MMMMMMMM	FFFFFFFF	BBBBBBBB

The relationship of the bitmask bits to the pixels in a cell is shown below.



Run Length Code

Runs of cells that use the same color can be described by a run length code, which is shown below. The code causes the next Count+1 cells to be rendered in the color that is specified in the Color0 byte.

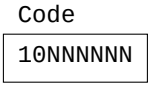


Escape Codes

A Cell encoder can use the following escape codes to perform operations such as loading colormaps, skipping frames, and including user data in the compressed bytestream. All escape codes start with a 1 in the code's most significant bit.

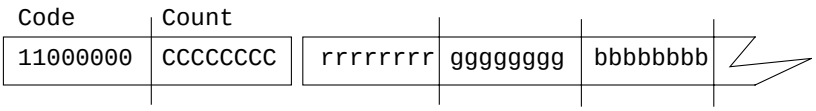
Skip Cells

The following code causes the next N+1 cells to be skipped (to a maximum of 64 cells).



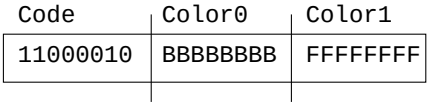
Load a New Colormap

The following code loads a new colormap. Count+1 byte-triples (red byte, green byte, and blue byte) follow the Count byte.



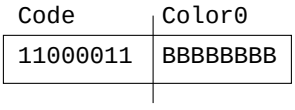
Use Same Mask with New Colors

The following code uses the same mask as the previous cell, but with new colors.



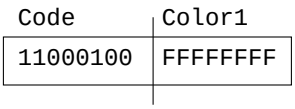
Use Same Mask with Color 0

The following code uses the same mask as the previous cell, but with a new color as indexed by the value of the Color0 byte.



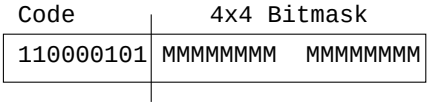
Use Same Mask with Color 1

The following code uses the same mask as the previous cell, but with a new color as indexed by the value of the Color1 byte.



Use Same Colors with New Mask

The following code uses the same colors as the previous cell, but with a new bitmask.



Include User Data

The following code enables you to include your own data in the bytestream. The next N+1 bytes in the bytestream are user data.

Skip an Entire Frame

The following code causes the next frame to be skipped.

Code
11000111

First Byte of Key-Frame Header

The following code is the first byte of a key-frame header, which was described earlier.

Code
11111111

Reserved Codes

The codes in the range 11001000 to 11111110 (inclusive) are reserved.

Summary of Cell Codes

Table D-1 lists the Cell codes.

Table D-1 Cell Bytestream Codes

Code	Description
0MMMMMMM MMMMMMMM FFFFFFFF BBBBBBBB	Cell code: M=bit mask, F=color 1 index, B=color 0 index
00000000 00000000 BBBBBBBB CCCCCCCC	Run length code: B=color 0 index, cell count = C+1
10NNNNNN	Interframe skip: cell count N+1
11000000 CCCCCCCC	Load new colormap: count of entries (RGB triples) that follow = C+1

Table D-1 Cell Bytestream Codes

Code	Description
11000010 BBBBBBBB FFFFFFFF	Same mask, new colors: B=color 0 index, F=color 1 index
11000011 BBBBBBBB	Same mask, new color 0: B=color 0 index
11000100 FFFFFFFF	Same mask, new color 1: F=color 1 index
11000101 0MMMMMMM MMMMMMMM	Same colors, new mask: M=bit mask
11000110 NNNNNNNN NNNNNNNN NNNNNNNN	User data: count of user bytes that follow = C+1
11001000-11111111	Reserved
11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111	Start of key frame header
00000001 WWWWWWWW WWWWWWWW HHHHHHHH HHHHHHHH	Key frame, image dimensions: W=width, H=height
00000010 RRRRRRRR RRRRRRRR RRRRRRRR RRRRRRRR	Key frame, frame rate in usec: R=frame rate
00000011 CCCCCCCC	Key frame, maximum colormap size = C+1
00000000	End of key frame header

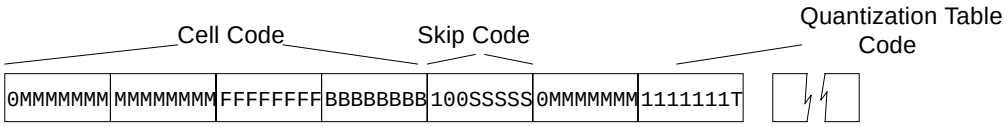
CellB Bytestream Description

CellB compression is designed for use in videoconferencing applications. Rather than indexing into a colormap to determine pixel colors (as in Cell), CellB is designed to be used with vector quantization and dequantization techniques in the YC_bC_r color space.

The CellB bytestream contains no information about image size and frame rates. It's the responsibility of the videoconferencing application to provide this information.

The CellB bytestream consists of:

- Cell codes
- Skip codes
- Quantization-table specification codes



Cell Code

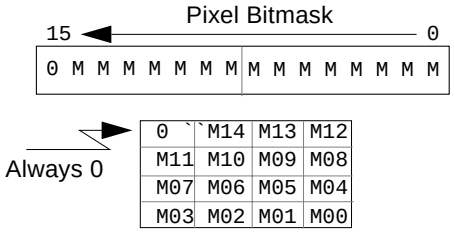
The four-byte code that describes a 4-by-4 CellB cell is shown below. There are two possible luminance (Y) levels for each cell but only one pair of chrominance (C_b and C_r) values.

The value in the C_b/C_r byte represents the chrominance component of the cell. The value in the Y/Y byte represents two luminance values (Y_0 and Y_1) that can represent the cell's luminance.

4x4 Bitmask	U/V Code	Y/Y Code
0MMMMMMM MMMMMMMM	UVUVUVUV	YYYYYYYY

If a pixel's bit is set in the 4-by-4 bitmask, the color that is rendered for the pixel is $<Y_1,C_b,C_r>$. If a pixel's bit is not set, the pixel is rendered by the color represented by $<Y_0,C_b,C_r>$.

The relationship of the bitmask bits to a cell's pixels is shown below.



C_b/C_r Quantization Table

The C_b/C_r field of the CellB bytecode represents the chrominance component of the cell. This C_b/C_r code is an index into a table of vectors that represent two independent components of chrominance. Figure D-3 on page 476 shows the default chrominance table. The section "Cb/Cr Table Values" on page 484 contains a list of the values used in the table.

The distribution of values in this default table is based on the observation that, in videoconferencing applications, a cell's C_b/C_r vectors are clustered around the origin (0,0). Therefore, the C_b/C_r codeword is circularly symmetric, with higher densities near the origin.

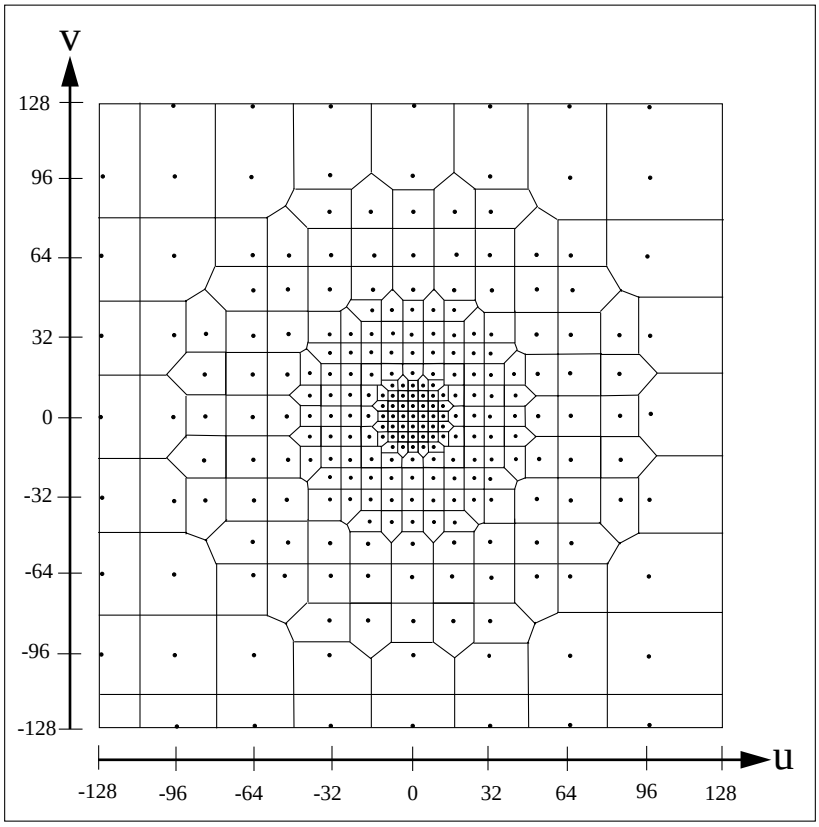


Figure D-3 Default CellB Chrominance Quantization Table

Y/Y Quantization Table

The Y/Y field of the CellB bytecode represents two luminance values of a cell (Y_0 and Y_1). This Y/Y code is an index into a table of two-component luminance vectors. Figure D-4 on page 478 shows the default luminance table. The section “Y/Y Table Values” on page 480 provides the list of values for the table.

The distribution of values in the default luminance table is statistically optimized. The quantizer takes advantage of the high correlation of luminance values within local regions of a cell. This results in a set of representative vectors that are most densely populated around the diagonal, where y1 equals y2.

An observer’s sensitivity to contrast is also taken into account, resulting in a distribution of points that is farther apart in regions where the contrast between two values is low.

In this example, the value for the Y/Y code is selected by approximating the mean luminance for the cell. Then, the pixels within the cell are separated into groups that are above and below the mean luminance. The mean luminance of these two groups is used to index a two-dimensional vector quantizer, which returns a byte value for the Y/Y code.

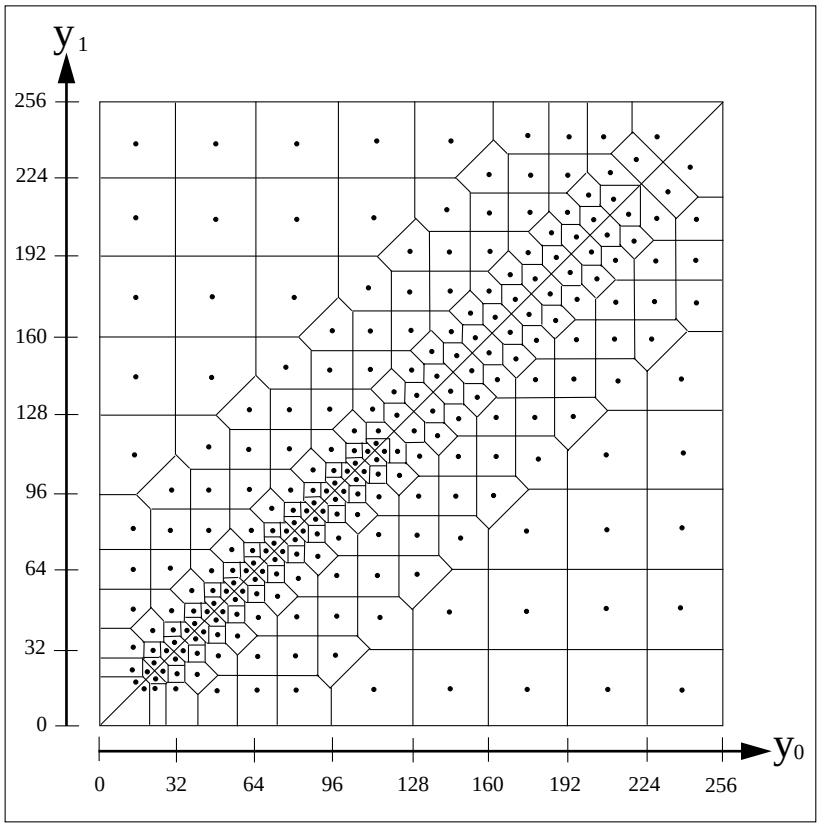


Figure D-4 Default CellB Luminance Quantization Table

Skip Code

The single-byte CellB skip code is shown below.

```
100SSSSS
```

The skip code tells the decoder to skip the next S+1 cells in the frame that is being decoded, which, for CellB, supports a simple form of interframe encoding. There are five skip bits, so that a maximum of 32 cells can be skipped with a single-byte skip code.

New Y/Y Table

The single-byte “new Y/Y table” code is shown below.

```
11111110
```

This code tells the decoder that the next 512 bytes are a new Y/Y quantization table. The bytes are arranged as:

Y1_000	Y2_000
Y1_001	Y2_001
.	.
.	.
.	.
Y1_255	Y2_255

Note – This code is not implemented in the current CellB compressor and decompressor.

New U/V Table

The single-byte “new U/V table” code is shown below.

```
11111111
```

This code tells the decoder that the next 512 bytes are a new U/V quantization table. The bytes are arranged as:

```
U_000    V_000
U_001    V_001
.        .
.        .
.        .
U_255    V_255
```

Note – This code is not implemented in the current CellB compressor and decompressor.

Default CellB Quantization Tables

Y/Y Table Values

Table D-2 lists the default values for the CellB Y/Y quantization table.

Table D-2 Default Y/Y Table *(1 of 4)*

Index	Y1	Y2	Index	Y1	Y2	Index	Y1	Y2
0	16	20	87	120	136	174	112	64
1	16	24	88	128	136	175	128	64
2	16	32	89	128	144	176	72	68
3	16	48	90	128	160	177	76	72
4	16	64	91	128	176	178	80	72
5	16	80	92	128	192	179	88	72

Table D-2 Default Y/Y Table *(2 of 4)*

Index	Y1	Y2	Index	Y1	Y2	Index	Y1	Y2
6	16	112	93	136	144	180	80	76
7	16	144	94	136	152	181	84	80
8	16	176	95	144	152	182	88	80
9	16	208	96	144	160	183	96	80
10	16	240	97	144	176	184	112	80
11	20	24	98	144	192	185	128	80
12	24	28	99	144	208	186	144	80
13	24	32	100	144	240	187	176	80
14	24	40	101	152	160	188	208	80
15	28	32	102	152	168	189	240	80
16	32	36	103	160	168	190	88	84
17	32	40	104	160	176	191	92	88
18	32	48	105	160	192	192	96	88
19	32	64	106	160	208	193	104	88
20	32	80	107	160	224	194	96	92
21	32	96	108	168	176	195	100	96
22	36	40	109	168	184	196	104	96
23	40	44	110	176	184	197	112	96
24	40	48	111	176	192	198	128	96
25	40	56	112	176	208	199	144	96
26	44	48	113	176	224	200	160	96
27	48	52	114	176	240	201	104	100
28	48	56	115	184	192	202	108	104
29	48	64	116	184	200	203	112	104
30	48	80	117	192	200	204	120	104
31	48	96	118	192	208	205	112	108
32	48	112	119	192	224	206	116	112

Table D-2 Default Y/Y Table *(3 of 4)*

Index	Y1	Y2	Index	Y1	Y2	Index	Y1	Y2
33	48	144	120	192	240	207	120	112
34	48	176	121	200	208	208	128	112
35	48	208	122	200	216	209	144	112
36	48	240	123	208	216	210	160	112
37	52	56	124	208	224	211	176	112
38	56	60	125	208	240	212	208	112
39	56	64	126	216	232	213	240	112
40	56	72	127	224	240	214	128	120
41	60	64	128	20	16	215	136	120
42	64	68	129	24	16	216	136	128
43	64	72	130	32	16	217	144	128
44	64	80	131	48	16	218	160	128
45	64	96	132	64	16	219	176	128
46	64	112	133	80	16	220	192	128
47	64	128	134	112	16	221	144	136
48	68	72	135	144	16	222	152	136
49	72	76	136	176	16	223	152	144
50	72	80	137	208	16	224	160	144
51	72	88	138	240	16	225	176	144
52	76	80	139	24	20	226	192	144
53	80	84	140	28	24	227	208	144
54	80	88	141	32	24	228	240	144
55	80	96	142	40	24	229	160	152
56	80	112	143	32	28	230	168	152
57	80	128	144	36	32	231	168	160
58	80	144	145	40	32	232	176	160
59	80	176	146	48	32	233	192	160

Table D-2 Default Y/Y Table *(4 of 4)*

Index	Y1	Y2	Index	Y1	Y2	Index	Y1	Y2
60	80	208	147	64	32	234	208	160
61	80	240	148	80	32	235	224	160
62	84	88	149	96	32	236	176	168
63	88	92	150	40	36	237	184	168
64	88	96	151	44	40	238	184	176
65	88	104	152	48	40	239	192	176
66	92	96	153	56	40	240	208	176
67	96	100	154	48	44	241	224	176
68	96	104	155	52	48	242	240	176
69	96	112	156	56	48	243	192	184
70	96	128	157	64	48	244	200	184
71	96	144	158	80	48	245	200	192
72	96	160	159	96	48	246	208	192
73	100	104	160	112	48	247	224	192
74	104	108	161	144	48	248	240	192
75	104	112	162	176	48	249	208	200
76	104	120	163	208	48	250	216	200
77	108	112	164	240	48	251	216	208
78	112	116	165	56	52	252	224	208
79	112	120	166	60	56	253	240	208
80	112	128	167	64	56	254	232	216
81	112	144	168	72	56	255	240	224
82	112	160	169	64	60			
83	112	176	170	68	64			
84	112	208	171	72	64			
85	112	240	172	80	64			
86	120	128	173	96	44			

C_b/C_r Table Values

Table D-3 lists the default values for the CellB C_b/C_r quantization table.

Table D-3 Default C_b/C_r Table *(1 of 4)*

Index	U	V	Index	U	V	Index	U	V
0	16	16	87	128	184	174	160	120
1	16	48	88	128	192	175	160	128
2	16	80	89	128	208	176	160	136
3	16	112	90	128	224	177	160	144
4	16	144	91	132	136	178	160	152
5	16	176	92	132	140	179	160	160
6	16	208	93	132	144	180	160	168
7	16	240	94	132	148	181	160	176
8	48	16	95	132	152	182	160	184
9	48	48	96	136	104	183	160	192
10	48	80	97	136	112	184	160	208
11	48	112	98	136	120	185	160	224
12	48	144	99	136	128	186	168	112
13	48	176	100	136	132	187	168	120
14	48	208	101	136	136	188	168	128
15	48	240	102	136	140	189	168	136
16	64	112	103	136	144	190	168	144
17	64	128	104	136	148	191	168	152
18	64	144	105	136	152	192	168	160
19	64	160	106	136	156	193	168	168
20	64	176	107	136	160	194	168	176
21	80	16	108	136	168	195	176	16
22	80	48	109	136	176	196	176	48
23	80	80	110	136	184	197	176	64
24	80	96	111	140	132	198	176	80

Table D-3 Default C_b / C_r Table *(2 of 4)*

Index	U	V	Index	U	V	Index	U	V
25	80	112	112	140	136	199	176	96
26	80	128	113	140	140	200	176	112
27	80	144	114	140	144	201	176	120
28	80	160	115	140	148	202	176	128
29	80	176	116	140	152	203	176	136
30	80	192	117	140	156	204	176	144
31	80	208	118	144	16	205	176	152
32	80	240	119	144	48	206	176	160
33	96	80	120	144	64	207	176	168
34	96	96	121	144	80	208	176	176
35	96	112	122	144	96	209	176	192
36	96	128	123	144	104	210	176	208
37	96	144	124	144	112	211	176	224
38	96	160	125	144	120	212	176	240
39	96	176	126	144	128	213	184	128
40	96	192	127	144	132	214	184	136
41	96	208	128	144	136	215	184	144
42	104	128	129	144	140	216	184	152
43	104	136	130	144	144	217	184	160
44	104	144	131	144	148	218	192	80
45	104	152	132	144	152	219	192	96
46	104	160	133	144	156	220	192	112
47	112	16	134	144	160	221	192	128
48	112	48	135	144	168	222	192	144
49	112	64	136	144	176	223	192	160
50	112	80	137	144	184	224	192	176
51	112	96	138	144	192	225	192	192

Table D-3 Default C_b / C_r Table *(3 of 4)*

Index	U	V	Index	U	V	Index	U	V
52	112	112	139	144	208	226	192	208
53	112	120	140	144	224	227	208	16
54	112	128	141	144	240	228	208	48
55	112	136	142	148	132	229	208	80
56	112	144	143	148	136	230	208	96
57	112	152	144	148	140	231	208	112
58	112	160	145	148	144	232	208	128
59	112	168	146	148	148	233	208	144
60	112	176	147	148	152	234	208	160
61	112	192	148	148	156	235	208	176
62	112	208	149	152	104	236	208	192
63	112	224	150	152	112	237	208	208
64	112	240	151	152	120	238	208	240
65	120	112	152	152	128	239	224	112
66	120	120	153	152	132	240	224	128
67	120	128	154	152	136	241	224	144
68	120	136	155	152	140	242	224	160
69	120	144	156	152	144	243	224	176
70	120	152	157	152	148	244	240	16
71	120	160	158	152	152	245	240	48
72	120	168	159	152	156	246	240	80
73	120	176	160	152	160	247	240	112
74	128	64	161	152	168	248	240	144
75	128	80	162	152	176	249	240	176
76	128	96	163	152	184	250	240	208
77	128	104	164	156	136	251	240	240
78	128	112	165	156	140	252	0	0

Table D-3 Default C_b / C_r Table *(4 of 4)*

Index	U	V	Index	U	V	Index	U	V
79	128	120	166	156	144	253	0	0
80	128	128	167	156	148	254	0	0
81	128	136	168	156	152	255	0	0
82	128	144	169	160	64			
83	128	152	170	160	80			
84	128	160	171	160	96			
85	128	168	172	160	104			
86	128	176	173	160	112			

D