

79	65	60	70	77	68	58	75
85	71	64	59	55	61	65	83
87	79	69	68	65	76	78	94

原图像包含 256 或 2^8 个可能的灰度级,因此,编码过程从对原子图像的像素层次移动 -2^7 或 -128 个灰度级开始。得到的移位阵列为:

-76	-73	-67	-62	-58	-67	-64	-55
-65	-69	-62	-38	-19	-43	-59	-56
-66	-69	-60	-15	16	-24	-62	-55
-65	-70	-57	-6	26	-22	-58	-59
-61	-67	-60	-24	-2	-40	-60	-58
-49	-63	-68	-58	-51	-65	-70	-53
-43	-57	-64	-69	-73	-67	-63	-45
-41	-49	-59	-60	-63	-52	-50	-34

对 $N=8$,根据式(8.5.24)和式(8.5.32)的正向 DCT,变换的阵列为:

-415	-29	-62	25	55	-20	-1	3
7	-21	-62	9	11	-7	-6	6
-46	8	77	-25	-30	10	7	-5
-50	13	35	-15	-9	6	0	3
11	-8	-13	-2	-1	1	-4	1
-10	1	3	-3	-1	0	2	-1
-4	-1	2	-1	2	-3	1	-2
-1	-1	-1	-2	-1	-1	0	-1

表 8.17 JPEG 系数编码分类

范 围	DC 差异类	AC 类
0	0	N/A
-1,1	1	1
-3,-2,2,3	2	2
-7, ..., -4,4, ..., 7	3	3
-15, ..., -8,8, ..., 15	4	4

(续表)

范 围	DC 差异类	AC 类
- 31, ..., - 16, 16, ..., 31	5	5
- 63, ..., - 32, 32, ..., 63	6	6
- 127, ..., - 64, 64, ..., 127	7	7
- 255, ..., - 128, 128, ..., 255	8	8
- 511, ..., - 256, 256, ..., 511	9	9
- 1 023, ..., - 512, 512, ..., 1 023	A	A
- 2 047, ..., - 1 024, 1 024, ..., 2 047	B	B
- 4 095, ..., - 2 048, 2 048, ..., 4 095	C	C
- 8 191, ..., - 4 096, 4 096, ..., 8 191	D	D
- 16 383, ..., - 8 192, 8 192, ..., 16 383	E	E
- 32 767, ..., - 16 384, 16 384, ..., 32 767	F	N/A

表 8.18 JPEG 默认 DC 编码(亮度)

分 类	基 础 编 码	长 度	分 类	基 础 编 码	长 度
0	010	3	6	1110	10
1	011	4	7	11110	12
2	100	5	8	111110	14
3	00	5	9	1111110	16
4	101	7	A	11111110	18
5	110	8	B	111111110	20

表 8.19 JPEG 默认 AC 编码(亮度)

扫描段/类	基 础 编 码	长 度	扫描段/类	基 础 编 码	长 度
0/0	1010(= EOB)	4			
0/1	00	3	8/1	11111010	9
0/2	01	4	8/2	11111111000000	17
0/3	100	6	8/3	111111110110111	19
0/4	1011	8	8/4	111111110111000	20
0/5	11010	10	8/5	111111110111001	21
0/6	111000	12	8/6	111111110111010	22
0/7	1111000	14	8/7	111111110111011	23
0/8	111110110	18	8/8	111111110111100	24
0/9	11111111000010	25	8/9	111111110111101	25
0/A	111111110000011	26	8/A	111111110111110	26
1/1	1100	5	9/1	111111000	10
1/2	111001	8	9/2	111111110111111	18
1/3	1111001	10	9/3	111111111000000	19
1/4	111110110	13	9/4	111111111000001	20
1/5	11111110110	16	9/5	111111111000010	21
1/6	111111110000100	22	9/6	111111111000011	22

(续表)

扫描段/类	基础编码	长度	扫描段/类	基础编码	长度
1/7	111111110000101	23	9/7	111111111000100	23
1/8	111111110000110	24	9/8	111111111000101	24
1/9	111111110000111	25	9/9	111111111000110	25
1/A	111111110001000	26	9/A	111111111000111	26
2/1	11011	6	A/1	11111001	10
2/2	11111000	10	A/2	111111111001000	18
2/3	111110111	13	A/3	111111111001001	19
2/4	111111110001001	20	A/4	111111111001010	20
2/5	111111110001010	21	A/5	111111111001011	21
2/6	111111110001011	22	A/6	111111111001100	22
2/7	111111110001100	23	A/7	111111111001101	23
2/8	111111110001101	24	A/8	111111111001110	24
2/9	111111110001110	25	A/9	111111111001111	25
2/A	111111110001111	26	A/A	111111111010000	26
3/1	111010	7	B/1	11111010	10
3/2	11110111	11	B/2	111111111010001	18
3/3	1111110111	14	B/3	111111111010010	19
3/4	111111110010000	20	B/4	111111111010011	20
3/5	111111110010001	21	B/5	111111111010100	21
3/6	111111110010010	22	B/6	111111111010101	22
3/7	111111110010011	23	B/7	111111111010110	23
3/8	111111110010100	24	B/8	111111111010111	24
3/9	111111110010101	25	B/9	111111111011000	25
3/A	111111110010110	26	B/A	111111111011001	26
4/1	111011	7	C/1	111111010	11
4/2	111111000	12	C/2	111111111011010	18
4/3	111111110010111	19	C/3	111111111011011	19
4/4	111111110011000	20	C/4	111111111011100	20
4/5	111111110011001	21	C/5	111111111011101	21
4/6	111111110011010	22	C/6	111111111011110	22
4/7	111111110011011	23	C/7	111111111011111	23
4/8	111111110011100	24	C/8	111111111100000	24
4/9	111111110011101	25	C/9	111111111100001	25
4/A	111111110011110	26	C/A	111111111100010	26
5/1	1111010	8	D/1	1111111010	12
5/2	111111001	12	D/2	111111111100011	18

(续表)

扫描段/类	基础编码	长度	扫描段/类	基础编码	长度
5/3	111111110011111	19	D/3	111111111100100	19
5/4	1111111110100000	20	D/4	111111111100101	20
5/5	1111111110100001	21	D/5	111111111100110	21
5/6	1111111110100010	22	D/6	111111111100111	22
5/7	1111111110100011	23	D/7	111111111101000	23
5/8	1111111110100100	24	D/8	111111111101001	24
5/9	1111111110100101	25	D/9	111111111101010	25
5/A	1111111110100110	26	D/A	111111111101011	26
6/1	1111011	8	E/1	11111110110	13
6/2	1111111000	13	E/2	11111111101100	18
6/3	111111110100111	19	E/3	11111111101101	19
6/4	111111110101000	20	E/4	111111111101110	20
6/5	111111110101001	21	E/5	111111111101111	21
6/6	111111110101010	22	E/6	111111111110000	22
6/7	111111110101011	23	E/7	111111111110001	23
6/8	111111110101100	24	E/8	111111111110010	24
6/9	111111110101101	25	E/9	111111111110011	25
6/A	111111110101110	26	E/A	111111111110100	26
7/1	11111001	9	F/0	121111110111	12
7/2	1111111001	13	F/1	11111111110101	17
7/3	111111110101111	19	F/2	111111111101110	18
7/4	1111111110110000	20	F/3	111111111110111	19
7/5	1111111110110001	21	F/4	111111111111000	20
7/6	1111111110110010	22	F/5	111111111111001	21
7/7	1111111110110011	23	F/6	111111111111010	22
7/8	1111111110110100	24	F/7	111111111111011	23
7/9	1111111110110101	25	F/8	111111111111100	24
7/A	1111111110110110	26	F/9	111111111111101	25
			F/A	111111111111110	26

如果用图 8.37(b)JPEG 推荐的标准化阵列对变换阵列进行量化,则进行按比例舍入[即,根据式(8.5.40)进行标准化]后的系数是:

-26 -3 -6 2 2 0 0 0

1	-2	-4	0	0	0	0	0
-3	1	5	-1	-1	0	0	0
-4	1	2	-1	0	0	0	0
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

这里,DC系数使用下式计算:

$$\begin{aligned}\hat{T}(0,0) &= \text{round}\left[\frac{T(0,0)}{Z(0,0)}\right] \\ &= \text{round}\left[\frac{-415}{16}\right] = -26\end{aligned}$$

注意,变换和标准化处理过程产生了大量的零值系数。当根据图 8.36(d)的 z 形模式对这些系数进行重新排列的时候,得到的一维系数序列是:

[-26 -3 1 -3 -2 -6 2 -4 1 -4 1 5 0 2 0 0 -1 2 0 0 0 0 0 -1 -1 EOB]

这里,EOB 字符代表块结尾标志。使用的专用 EOB 霍夫曼码字(见表 8.19 中的类 0 和行程长度 0)表明在重新排列的序列中系数余项均为 0。

对于重排列系数序列默认的 JPEG 编码的构造,是从计算当前的 DC 系数和前面经过编码的子图像的 DC 系数之间的差开始的。由于这里使用的子图像来自图 8.23,并且紧挨着此子图像左边的经过变换和量化的子图像的 DC 系数值为 -17,所以得到的 DPCM 差为 $[-26 - (-17)]$ 或 -9,此值位于表 8.17 DC 差异类 4 中。根据表 8.18 的默认霍夫曼差异码,对一个类 4 的差的合理基础编码是 101(一个 3 比特编码),而经过完整编码的类 4 系数的总长度应有 7 比特。余下的 4 比特应该根据差值的最低有效位(LSB)生成。对一个一般的 DC 差异类(如,类 K),需要额外的 K 比特,并将正差或负差的 K 个 LSB 进行减 1 运算。对差为 -9 的值,合理的 LSB 为 (0111) - 1 或 0110,且完整的 DPCM 编码的 DC 码字是 1010110。

重排阵列的非零 AC 系数根据表 8.17 和表 8.19 可以进行类似的编码。主要差别在于每个默认的 AC 霍夫曼码字依赖于前述对非零系数编码后的零值系数的数目,以及非零系数的量级分类(见表 8.19 中的标记为扫描段/类的列)。因此,第一个重排阵列的非零 AC 系数(-3)的编码为 0100。这个编码的前两位表明这个系数属于量级分类 2 且没有零值系数在它的前面(见表 8.17);最后两位的生成使用了与获得 DC 差异编码的 LSB 所使用的相同处理过程。继续使用这种方法,完整的编码(重排列)阵列为:

1010110 0100 001 0100 0101 100001 0110 100011 001 100011 001
001 100101 11100110 110110 0110 11110100 000 1010

这里分别插入了空格以便于阅读。尽管在这个例子中并不需要这个阵列,但默认的 JPEG

编码包含对一个0后面跟着15个0的段设立的专用码字(见表8.19中的类0和行程长度F)。在完全编过码的重排阵列中的总比特数(在这个例子中,表示整幅 8×8 大小的8位子图像所要求的比特数)是92。得到的压缩率为 $512/92$,或大约5.6:1。

为了解压缩一幅压缩后的JPEG子图像,解码器必须首先重新产生标准化的导出压缩比特流的变换系数。因为一个霍夫曼编码的二值序列具有瞬时性并且其解码具有惟一性,所以用简单的查询表方式就可以很容易地实现这一步骤。这里量化系数的重新生成的阵列为:

-26	-3	-6	2	2	0	0	0
1	-2	-4	0	0	0	0	0
-3	1	5	-1	-1	0	0	0
-4	1	2	-1	0	0	0	0
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

在根据式(8.5.42)进行反标准化之后,阵列变成:

-416	-33	-60	32	48	0	0	0
12	-24	-56	0	0	0	0	0
-42	13	80	-24	-40	0	0	0
-56	17	44	-29	0	0	0	0
18	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

这里,例如DC系数,可以根据下式计算:

$$\hat{T}(0,0) = \hat{T}(0,0)Z(0,0) = (-26)(16) = -416$$

完整的重构子图像通过式(8.5.25)和式(8.5.32)得到的反向标准化阵列的逆DCT得出:

-70	-64	-61	-64	-69	-66	-58	-50
-72	-73	-61	-39	-30	-40	-54	-59

-68	-78	-58	-9	13	-12	-48	-64
-59	-77	-57	0	22	-13	-51	-60
-54	-75	-64	-23	-13	-44	-63	-56
-52	-71	-72	-54	-54	-71	-71	-54
-45	-59	-70	-68	-67	-67	-61	-50
-35	-47	-61	-66	-60	-48	-44	-44

而对每个逆变换像素用 $+2^7$ (或 $+128$) 进行移位得到:

58	64	67	64	59	62	70	78
56	55	67	89	98	88	74	69
60	50	70	119	141	116	80	64
69	51	71	128	149	115	77	68
74	53	64	105	115	84	65	72
76	57	56	74	75	57	57	74
83	69	59	60	61	61	67	78
93	81	67	62	69	80	84	84

原子图像和重构子图像之间的任何差异都是 JPEG 压缩和解压缩处理过程固有的有损性造成的。在这个例子中,误差范围从 -14 到 $+11$, 并且呈如下的分布状态:

-6	-9	-6	2	11	-1	-6	-5
7	4	-1	1	11	-3	-5	3
2	9	-2	-6	-3	-12	-14	9
-6	7	0	-4	-5	-9	-7	1
-7	8	4	-1	11	4	3	-2
3	8	4	-4	2	11	1	1
2	2	5	-1	-6	0	-2	5
-6	-2	2	6	-4	-4	-6	10

整体压缩和重构处理的均方根误差接近 5.9 个灰度级。

前面例子中的重构子图像大约在以图 8.38(a)中妇女的右眼为中心的位置上。这里请注意,在原始子图像和重构子图像上,第 4 行第 5 列的位置上都含有一个局部灰度级的波峰,这

个位置是妇女的瞳孔反射的一个亮点。这个局部波峰造成重构子图像的均方根误差明显超过了完全解压缩图像的总误差。实际上,这个误差接近图 8.38(a)中误差的两倍。图 8.38(a)的图像也是使用基准 JPEG 算法压缩得到的。原因是,原图像中的许多子图像都接近恒定并且描述这些图像时几乎没有失真。图 8.38(b)给出了另外一个使用 JPEG 基准压缩得到的结果。

JPEG 2000

尽管这种标准仍未被正式采用,但 JPEG 2000 相对于原始的 JPEG 标准,在对连续色调静止图像的压缩方面以及对压缩数据的访问方面提供了更大的灵活性。例如,一幅按 JPEG 2000 标准压缩的图像的一些部分可以从图像中提取出来进行转发、存储、显示和/或编辑。这个标准是以 8.5.3 节的小波编码技术为基础的。系数量化与单一尺度和子带相适应,并且量化的系数以位平面为基础(见 8.4 节)进行算术编码。使用标准符号,一幅图像可以进行如下(ISO/IEC[2000])编码。

编码处理的第一步是通过减 $2^{S_{siz}-1}$ 将被编码的 S_{siz} 比特无符号图像样值向 DC 级移动。如果图像具有不止一个组成分量——比如彩色图像的红色、绿色和蓝色面——则对每个分量进行单独的移动。如果有 3 个精确的分量,则可以使用可逆的或非可逆的线性组合进行任意选择的解相关处理。标准中不可逆的分量变换是:

$$\begin{aligned} Y_0(x, y) &= 0.299I_0(x, y) + 0.587I_1(x, y) + 0.114I_2(x, y) \\ Y_1(x, y) &= -0.16875I_0(x, y) - 0.33126I_1(x, y) + 0.5I_2(x, y) \\ Y_2(x, y) &= 0.5I_0(x, y) - 0.41869I_1(x, y) - 0.08131I_2(x, y) \end{aligned} \quad (8.6.1)$$

这里 I_0, I_1 和 I_2 是级别的平移输入分量, Y_0, Y_1 和 Y_2 是相应的解相关分量。如果输入分量是彩色图像的红色、绿色和蓝色平面,式(8.6.1)用 $Y'CbCr$ 彩色视频变换来近似 $R'G'B'$ (Poynton [1996])^①。变换的目的是改善压缩效率;变换后的分量 Y_1 和 Y_2 是差值图像,此图像的直方图在零点附近具有很高的峰。

在图像进行了水平移动和选择性的解相关处理之后,它的分量被分割成像块。这些块是像素的矩形阵列,而这些像素包含着所有分量相同的相关部分。因此,像块处理生成了可以独立进行提取和重构的块分量,这种处理提供了一种对编码图像的某一有限区域访问和/或操作的简单原理。

对每个块分量的行和列的一维离散小波变换就可以计算了。对于无误差压缩,这种变换是以双正交、5-3 系数尺度和小波向量(Le Gall 和 Tabatabai[1988])为基础的。对于非整数值变换系数还定义了一个四舍五入的程序。在有损应用中,使用 9-7 小波向量尺度系数(Antonini, Barlaud, Mathieu 和 Daubechies[1992])。无论哪种情况,都是使用 7.4 节的快速小波变换或通过一种基于提升的方法进行计算的。表 8.20 中给出了构造一个 9-7 FWT 分析滤波器组所需的系数。补充的基于提升的实现涉及 6 种连续的“提升”和“尺度”操作:

① $R'G'B'$ 是线性 CIE(International Commission on Illumination, 国际照明委员会)RGB 色度值经图像矫正且非线性的版本。 Y 是亮度, C_b 和 C_r 是色差(即,标定的 $B' - Y$ 和 $R' - Y$ 值)。

$$\begin{aligned}
Y(2n+1) &= X(2n+1) + \alpha[X(2n) + X(2n+2)], i_0 - 3 \leq 2n+1 < i_1 + 3 \\
Y(2n) &= X(2n) + \beta[Y(2n-1) + Y(2n+1)], i_0 - 2 \leq 2n < i_1 + 2 \\
Y(2n+1) &= Y(2n+1) + \gamma[Y(2n) + Y(2n+2)], i_0 - 1 \leq 2n+1 < i_1 + 1 \quad (8.6.2) \\
Y(2n) &= Y(2n) + \delta[Y(2n-1) + Y(2n+1)], i_0 \leq 2n < i_1 \\
Y(2n+1) &= -K \cdot Y(2n+1), i_0 \leq 2n+1 < i_1 \\
Y(2n) &= Y(2n)/K, i_0 \leq 2n < i_1
\end{aligned}$$

这里, X 是被变换的块分量, Y 是变换结果, i_0 和 i_1 定义了分量内的块分量的位置。即, 它们是将要变换的块分量行和列的第一个取样及紧接着的后一个取样的索引。变量 n 是以 i_0, i_1 为基础计算的, 并且将形成 6 种操作。如果 $n < i_0$ 或 $n \geq i_1$, 则 $X(n)$ 通过对称地对 X 进行扩展而得到。例如, $X(i_0 - 1) = X(i_0 + 1), X(i_0 - 2) = X(i_0 + 2), X(i_1) = X(i_1 - 2), X(i_1 + 1) = X(i_1 - 3)$ 。在关于提升和尺度操作的总结部分, Y 的偶下标的值与 FWT 低通滤波输出相等; Y 的奇数下标的值与高通 FWT 滤波输出相对应。提升参量 α, β, γ 和 δ 分别为 $-1.586134342, -0.052980118, 0.882911075$ 和 0.433506852 。尺度因子 K 为 1.230174105 。

表 8.20 不可逆的 9-7 小波变换的低通和高通分析滤波器的脉冲响应

滤波器抽头	高通小波系数	低通尺度系数
0	-1.115087052456994	0.6029490182363579
± 1	0.5912717631142470	0.2668641184428723
± 2	0.05754352622849957	-0.07822326652898785
± 3	-0.09127176311424948	-0.01686411844287495
± 4	0	0.02674875741080976

刚才描述的变换生成了 4 个子带——分别是块分量的低分辨率近似和分量的水平、垂直及对角频率特征。后续迭代限制前面分解的近似值系数, 将这种变换重复 N_L 次可以得到一个 N_L 尺度的小波变换。相邻尺度的变换联系到 2 的幂次空间且最低尺度仅包含显式定义的原始块分量的近似。正如在图 8.46 中推测的那样, 这里总结了 $N_L = 2$ 时的标准符号, 一个一般的 N_L 尺度变换包含了 $3N_L + 1$ 个子带, 而这些子带系数, 对于 $b = N_L LL, N_L HL, \dots, 1HL, 1LH, 1HH$, 用 a_b 代表。这个标准没有指明需要计算的尺度的数目。

当每个块分量都经过处理之后, 变换系数的总数等于原始图像中的取样数目, 但重要的可视信息被集中于少数系数中。为了减少表示变换的数据比特数, 子带 b 的系数 $a_b(u, v)$ 使用下式量化为值 $q_b(u, v)$:

$$q_b(u, v) = \text{sign}[a_b(u, v)] \cdot \text{floor}\left[\frac{|a_b(u, v)|}{\Delta_b}\right] \quad (8.6.3)$$

这里量化步长 Δ_b 为:

$$\Delta_b = 2^{R_b - \epsilon_b} \left(1 + \frac{\mu_b}{2^{11}}\right) \quad (8.6.4)$$

R_b 是子带 b 的标定动态范围, 而 ϵ_b 和 μ_b 是分配给子带系数的指数和尾数的比特数。子带 b 的

标定动态范围是,用于表示原始图像的比特数和对子带 b 的分析增益比特的位数之和。子带分析增益比特遵从图 8.46 中显示的简单模式。例如,对子带 $b = 1HH$ 存在两个分析增益比特。

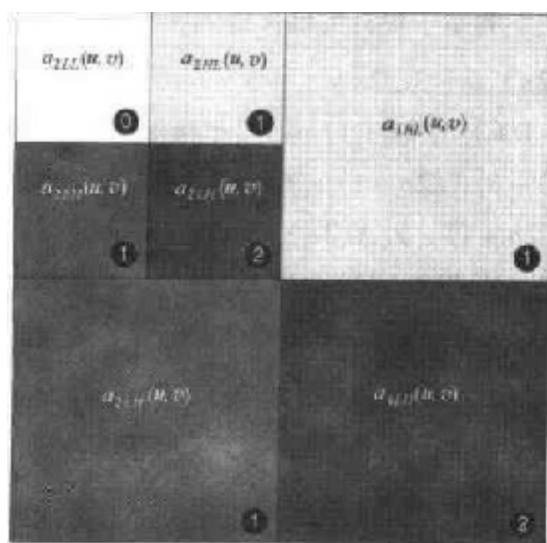


图 8.46 JPEG 2000 的 2 尺度小波变换块分量系数符号和分析结果

对无误差压缩, $\mu_b = 0$, $R_b = \epsilon_b$, $\Delta_b = 1$ 。对不可逆压缩,在标准中没有指定特定的量化步长。相反,必须以子带为基础向解码器提供指数和尾数的比特数,这被称为显式量化,或仅对 N_{iLL} 子带,称为隐式量化。对于后一种情况,对余下的子带使用推测的 N_{iLL} 子带参数进行量化。令 ϵ_0 和 μ_0 表示分配到 N_{iLL} 子带的位数,对子带 b 推测的参数为:

$$\begin{aligned} \mu_b &= \mu_0 \\ \epsilon_b &= \epsilon_0 + nsd_b - nsd_0 \end{aligned} \quad (8.6.5)$$

这里, nsd_b 表示由原始图像块分量到子带 b 的子带分解层次的数目。

编码过程的最后一步是系数比特建模、算术编码、比特流分层和分组。每个变换块分量的子带系数排列在称为码块的矩形块中,这个块一次对一个位平面进行独立编码。从带有非零元素的最高有效位平面开始,每个位平面进行 3 次处理。位平面的每比特仅在 3 次处理中的一次进行编码。其过程分别称为有效传播、量级细化和净化。然后,对输出进行算术编码并从其他编码块使用相似的途径进行分组以形成层。一个层是来自每个编码块的编码途径分组的任意数。得到的层被最终分割成信息包,提供从总体编码流中提取某一空间区域的附加方法。包是编码码流的基本单元。

JPEG 2000 解码器是前述过程的简单反操作。对比特建模、算术编码、分层和编码流打包进行解码之后,对原始图像块分量的用户选择的数重建子带。尽管编码器可能已经在特定的子带上对 M_b 位平面进行了编码,但用户可以选择只用 N_b 位平面进行解码,这取决于嵌入的编码流的性质。这实际上是使用 $2^{M_b - N_b} \cdot \Delta_b$ 的步长对编码块系数进行量化。所有未解码的比特都设置为零,并且对得到的表示为 $\bar{q}_b(u, v)$ 的系数使用下式进行反量化:

$$R_{q_b}(u, v) = \begin{cases} (\bar{q}_b(u, v) + 2^{M_b - N_b(u, v)} \cdot \Delta_b) & \bar{q}_b(u, v) > 0 \\ (\bar{q}_b(u, v) - 2^{M_b - N_b(u, v)} \cdot \Delta_b) & \bar{q}_b(u, v) < 0 \\ 0 & \bar{q}_b(u, v) = 0 \end{cases} \quad (8.6.6)$$