ECE462 – Lecture 12

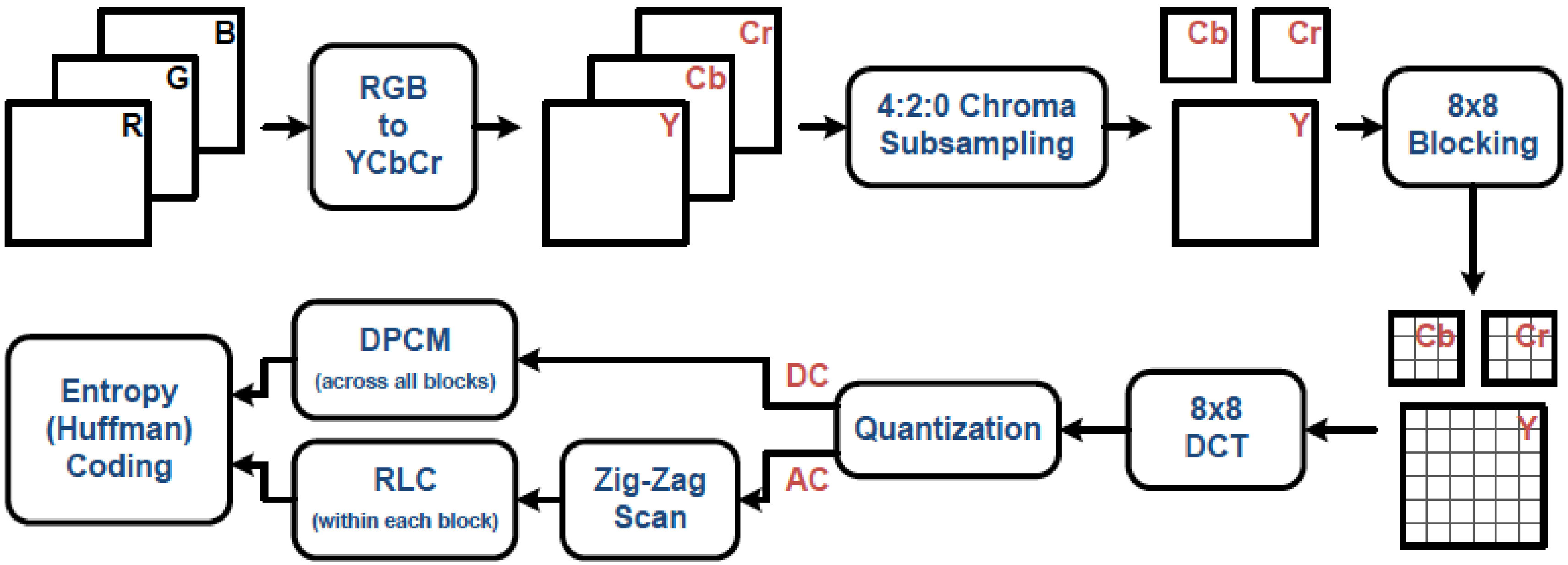
JPEG – DCT Based Image Compression

JPEG Standard Overview

- "Joint Photographic Experts Group" (JPEG) formally accepted international image compression standard in 1992
- Transform-based lossy compression scheme with the following core components
  - RGB $\rightarrow$ YCbCr colour transform
  - Subsample chrominance channels (4:2:0)
  - 8x8 DCT applied to non-overlapping blocks
  - Uniform midtread scalar quantization applied to DCT coefficients
  - Coefficient coding
    - DC coefficients DPCM coded
    - AC coefficients zig-zag ordered and run-length coded (RLC)
  - Entropy (Huffman) coding of both DPCM coded DC coefficients and RLC coded AC coefficients.

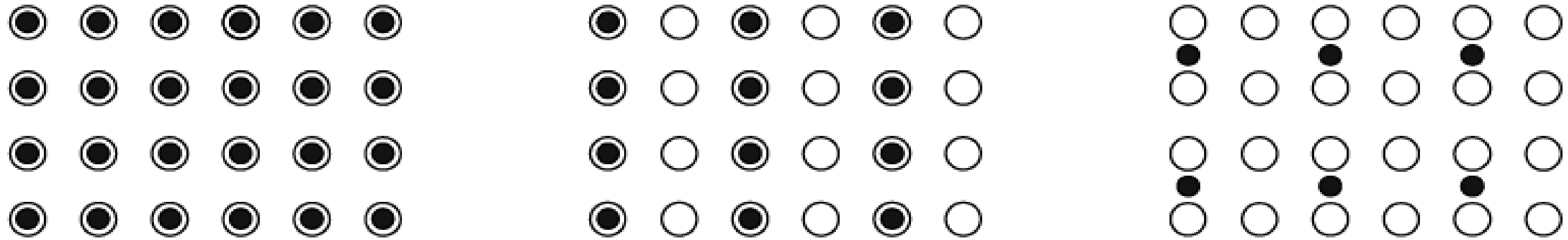
JPEG Standard Overview (cont'd)

- Sequential mode (default)
  - Other modes (progressive, hierarchical and lossless) not common and not covered in this course



Chroma Subsampling

- Chroma subsampling: reduces spatial resolution of chrominance channels (see text 5.3.1)
  - Human visual system is less sensitive to chrominance resolution
- Upon reconstruction, missing chrominance pixels must be interpolated via filtering



4:4:4 | 4:2:2 | 4:2:0

-No chroma subsampling

-Chroma subsampled by factor of 2 horizontally

-Chroma subsampled by factor of 2 horizontally and vertically

-Vertical filtering results in chroma pixels positioned between luminance pixels

-Used in JPEG and MPEG

○ Pixel with only Y value

● Pixel with only Cr and Cb values

◉ Pixel with Y, Cr, and Cb values

Blocking and DCT

- Each channel is divided into 8x8 non-overlapping blocks
  - Represents "spatial range" of correlation in a typical photographic image
  - Larger blocks would introduce pixels that are less correlated (e.g., from opposing edges of the block) and would result in less energy compaction after transform
- 8x8 DCT is applied independently to each block
  - Since luminance pixels have range [0,255], 128 is subtracted before DCT is applied in order to make values approximately zero-mean
    - DC coefficients will have generally lower magnitude
    - AC coefficients unaffected
  - Each block has 1 DC coefficient and 63 AC coefficients

Quantization

- Uniform midtread scalar quantization applied to all coefficients

$$\hat{F}(u,v) = \mathrm{round}\big(F(u,v)/Q(u,v)\big) \quad \text{where } u,v = 0\ldots7$$

$$\text{Reverse (scaling)}: \quad \tilde{F}(u,v) = \hat{F}(u,v)\cdot Q(u,v)$$

- Quantization is main source of loss in JPEG
- Different 8x8 quantization table (or “Q table”) for luminance and chrominance channels
  - Generally, larger step sizes (heavier quantization) for chrominance
- Values in table are quantizer step size
- Smaller step sizes for low frequency components, larger step sizes for high frequency
  - Human visual system is less sensitive to high spatial frequencies
  - Image content generally varies slowly (contains mostly low spatial frequencies)

Quantization (cont'd)

- JPEG standard recommended Q tables:

Q_Y

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Q_{Ch}

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

- Rate vs. distortion can be varied by scaling the Q tables
  - E.g.: $\text{Scale factor} = e^{\left(12 \cdot \frac{100-\text{quality}}{100}\right)}$
  - Not specified in the JPEG standard
- When heavy quantization is applied (low quality), block edges become visible (because of discontinuity between blocks), producing "blocking artifacts"

Quantization (cont'd)

An 8×8 block from the Y image of 'Lena'

200	202	189	188	189	175	175	175
200	203	198	188	189	182	178	175
203	200	200	195	200	187	185	175
200	200	200	200	197	187	187	187
200	205	200	200	195	188	187	175
200	200	200	200	200	190	187	175
205	200	199	200	191	187	187	175
210	200	200	200	188	185	187	186

$f(i, j)$

Pixel Block (smooth region)

→ **DCT**

515	65	-12	4	1	2	-8	5
-16	3	2	0	0	-11	-2	3
-12	6	11	-1	3	0	1	-2
-8	3	-4	2	-2	-3	-5	-2
0	-2	7	-5	4	0	-1	-4
0	-3	-1	0	4	1	-1	0
3	-2	-3	3	3	-1	-1	3
-2	5	-2	4	-2	2	-3	0

$F(u, v)$

DCT Coefficients

Quantization (cont'd)

Quantize

Quantized Coefficients

32	6	-1	0	0	0	0	0
-1	0	0	0	0	0	0	0
-1	0	1	0	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
0	0	0	0	0	0	0	0

$\hat{F}(u,v)$

Rev. Quantize

Reverse Quantized (scaled)

512	66	-10	0	0	0	0	0
-12	0	0	0	0	0	0	0
-14	0	16	0	0	0	0	0
-14	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

$\tilde{F}(u,v)$

Inverse DCT

199	196	191	186	182	178	177	176
201	199	196	192	188	183	180	178
203	203	202	200	195	189	183	180
202	203	204	203	198	191	183	179
200	201	202	201	196	189	182	177
200	200	199	197	192	186	181	177
204	202	199	195	190	186	183	181
207	204	200	194	190	187	185	184

$\tilde{f}(i,j)$

Reconstructed Pixel Block

1	6	-2	2	7	-3	-2	-1
-1	4	2	-4	1	-1	-2	-3
0	-3	-2	-5	5	-2	2	-5
-2	-3	-4	-3	-1	-4	4	8
0	4	-2	-1	-1	-1	5	-2
0	0	1	3	8	4	6	-2
1	-2	0	5	1	1	4	-6
3	-4	0	6	-2	-2	2	2

$\epsilon(i,j) = f(i,j) - \tilde{f}(i,j)$

Reconstruction Error

Quantization (cont'd)

Another 8 × 8 block from the Y image of 'Lena'

70	70	100	70	87	87	150	187
85	100	96	79	87	154	87	113
100	85	116	79	70	87	86	196
136	69	87	200	79	71	117	96
161	70	87	200	103	71	96	113
161	123	147	133	113	113	85	161
146	147	175	100	103	103	163	187
156	146	189	70	113	161	163	197

$f(i,j)$

Pixel Block (textured region)

→ **DCT**

-80	-40	89	-73	44	32	53	-3
-135	-59	-26	6	14	-3	-13	-28
47	-76	66	-3	-108	-78	33	59
-2	10	-18	0	33	11	-21	1
-1	-9	-22	8	32	65	-36	-1
5	-20	28	-46	3	24	-30	24
6	-20	37	-28	12	-35	33	17
-5	-23	33	-30	17	-5	-4	20

$F(u,v)$

DCT Coefficients

Quantization (cont'd)

Quantize →

Quantized Coefficients

-5	-4	9	-5	2	1	1	0
-11	-5	-2	0	1	0	0	-1
3	-6	4	0	-3	-1	0	1
0	1	-1	0	1	0	0	0
0	0	-1	0	0	1	0	0
0	-1	1	-1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

$\hat{F}(u,v)$

Rev. Quantize →

Reverse Quantized (scaled)

-80	-44	90	-80	48	40	51	0
-132	-60	-28	0	26	0	0	-55
42	-78	64	0	-120	-57	0	56
0	17	-22	0	51	0	0	0
0	0	-37	0	0	109	0	0
0	-35	55	-64	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

$\tilde{F}(u,v)$

Inverse DCT →

70	60	106	94	62	103	146	176
85	101	85	75	102	127	93	144
98	99	92	102	74	98	89	167
132	53	111	180	55	70	106	145
173	57	114	207	111	89	84	90
164	123	131	135	133	92	85	162
141	159	169	73	106	101	149	224
150	141	195	79	107	147	210	153

$\tilde{f}(i,j)$

Reconstructed Pixel Block

0	10	-6	-24	25	-16	4	11
0	-1	11	4	-15	27	-6	-31
2	-14	24	-23	-4	-11	-3	29
4	16	-24	20	24	1	11	-49
-12	13	-27	-7	-8	-18	12	23
-3	0	16	-2	-20	21	0	-1
5	-12	6	27	-3	2	14	-37
6	5	-6	-9	6	14	-47	44

$\epsilon(i,j) = f(i,j) - \tilde{f}(i,j)$

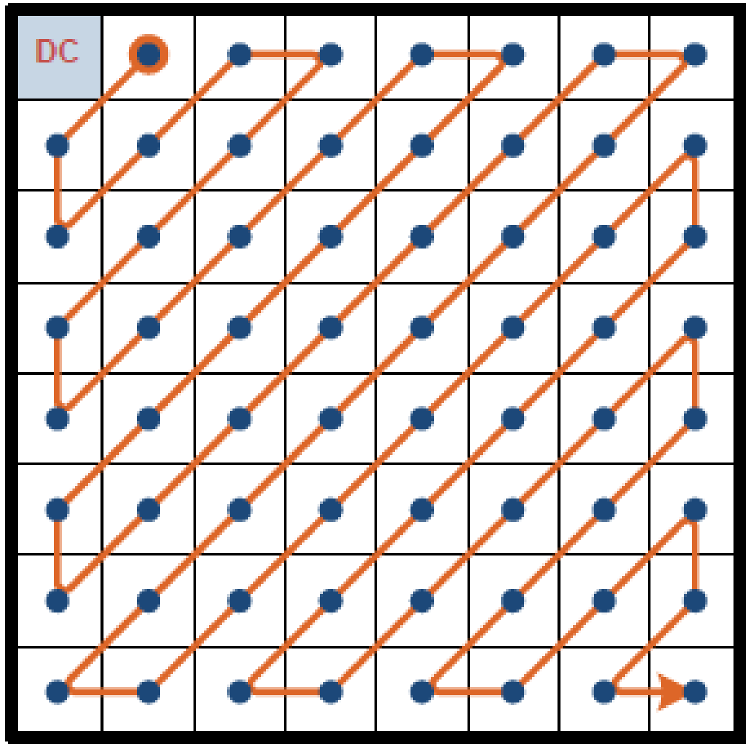
Reconstruction Error

Coefficient Coding

- DPCM is applied to all the DC coefficients
  - Predictor is (one) previous coefficient, without weighting
  - No quantization applied in this step
  - Ex.: The DC coefficients for the first 5 blocks:
    DC coefficients: [150,155,149,152,144]
    After DPCM: [150,5,-6,3,-8]
  - Since the DC coefficients represent the average value for their respective 8x8 blocks, coding them together in series using DPCM takes advantage of very slowly varying luminance or colour

Coefficient Coding (cont'd)

- AC coefficients are zig-zag scanned and run-length coded (RLC)
  - Zig-zag scan takes the 63 AC coefficients and orders them in a vector from low to high frequency



Coefficient Coding (cont'd)

- Zig-Zag example (smooth block from "Lena"):
  [6,-1,-1,0,-1,0,0,0,-1,0,0,1,0,0,…,0]
- Run-length coding (RLC) replaces zig-zag vector with a sequence of (`run-length`,`value`) pairs
  - `run-length` is the number of zeros before `value`
  - Special code (0,0) indicates end of block – remainder of vector is all zeros
- Output sequence from vector above:
  (0,6)(0,-1)(0,-1)(1,-1)(3,-1)(2,1)(0,0)

Entropy (Huffman) Coding

- Huffman coding is applied to both DC and AC coefficients to generate binary output
- DPCM-coded DC coefficients are represented as (`size`,`amplitude`) pairs, where `size` is the number of bits needed to represent the coefficient value, and `amplitude` is the actual coefficient value (1's complement)
  - Continuing previous example – 5 DPCM-coded DC coefficients: [150,5,-6,3,-8]
  - Output pairs: (8,10010110)(3,101)(3,001)(2,11)(4,0111)
  - `size` is Huffman coded since small values occur frequently
  - `amplitude` is output directly since the values vary widely

Entropy (Huffman) Coding (cont'd)

- RLC-coded AC coefficients, represented as (`run-length`,`value`) sequences have `value` represented using (`size`,`amplitude`) (as with DC)
- `run-length` and `size` each represented using 4 bits, concatenated (into 1 byte) and treated as a symbol for Huffman coding
- `amplitude` bits are output directly (as with DC)
- Since `run-length` is limited to 4 bits, only run-lengths of up to 15 can be represented
  - Special extension code inserted for longer run-lengths

Example Output

- Output of lossy compression schemes generally evaluated using rate vs. distortion plots
  - Rate (or bit-rate) is average number of bits per pixel (bpp) in coded output – calculated as (total file size)/(# of pixels)
  - Distortion is typically measured using PSNR
  - Compression ratio can be calculated by taking the “raw” bit-rate and dividing by the output bit-rate
    - Example: RGB image size = 512x512 pixels
      compressed output file size = 49152 bytes;
      ∴ output bit-rate = 49152*8/(512*512) = 1.5 bpp
      compression ratio = 24 bpp / 1.5 bpp = 16:1

Example Output (cont'd)

Original Image

JPEG ("Quality"=1)

PSNR = 19.3 dB
Bit-rate = 0.23 bpp
Comp. Ratio = 107:1

Example Output (cont'd)

Original Image

JPEG ("Quality"=50)

PSNR = 29.3 dB
Bit-rate = 1.47 bpp
Comp. Ratio = 16:1

Example Output (cont'd)

Original Image

JPEG ("Quality"=75)

PSNR = 32.0 dB
Bit-rate = 2.15 bpp
Comp. Ratio = 11:1

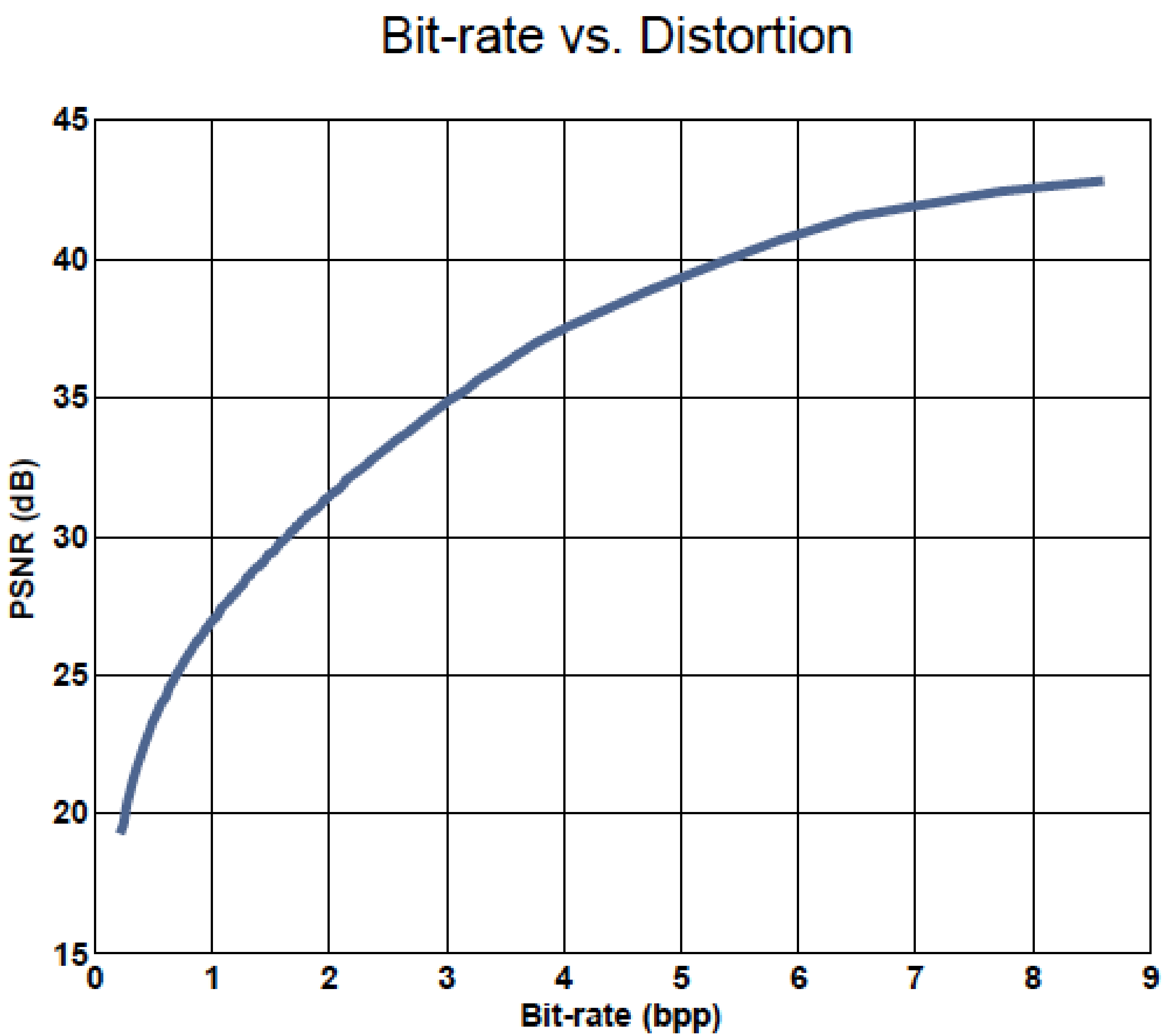
Example Output (cont'd)

Original Image

JPEG ("Quality"=100)

PSNR = 42.8 dB
Bit-rate = 8.60 bpp
Comp. Ratio = 3:1

Example Output (cont'd)



- Percentage indicates quality factor
- Note: The usual default quality factor is 75%.
- As quality factor decreases (higher compression) “blocking effects” become visible. (originating from the block DCT operation)