Lecture 6 – JPEG Image Compression

PORTLAND STATE UNIVERSITY

JPEG

Modes

- Sequential - encodes in "scanlines" left-to-right top-to-bottom
- Progressive - encode image in multiple coarse to clear passes
  - +) useful in bandwidth constrained environments

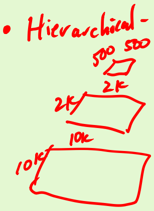
99% ↓ growing

20:1 compression

- Lossless - guarantees exact recovery of source image
  - −) much lower compression ratios

2.4:1

- Hierarchical - encodes image in multiple resolutions
  - +) access to smaller versions w/o decompressing whole thing



PORTLAND STATE UNIVERSITY

JPEG - Provides great flexibility in compression

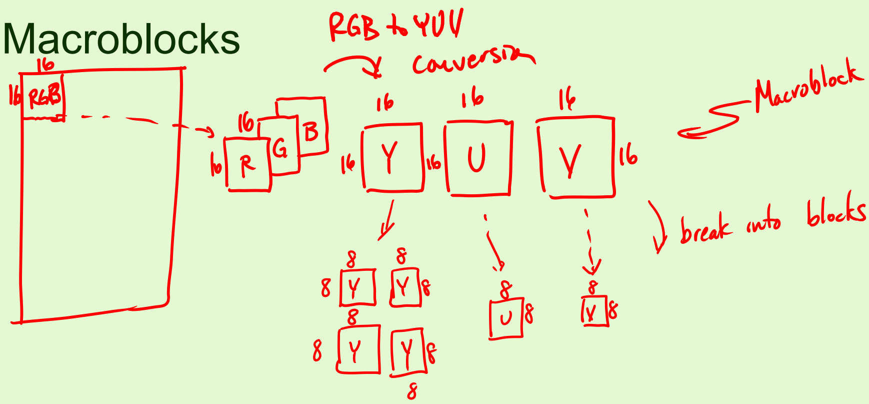
Quality knob that is exposed to user

+) Users can trade-off file size and quality of image

STEPS

- •) Break image into 16×16 pixel blocks "macroblocks"
  Using YUV format
- •) Break macroblock into blocks ← 8×8 pixels
- •) DCT = discrete cosine transform
  Transform from "spatial" domain to "frequency" domain
- •) Quantization ⇐ primary mechanism for quality control
- •) Entropy encoding
  → Differential encoding
  → Huffman
  → Run length encoding

Macroblocks



For each block do
DCT + Quant, entropy encoding

Basic Transform Coding

Amplitude 200

A tone = 440Hz (?)

Spatial domain

CD-audio Samples

16-bit samples @ 44,100Hz

frequency 440Hz @ 200

Fourier that said any signal can be decomposed into a summation of frequencies

f

1.5f

2f

2.5f

PORTLAND STATE UNIVERSITY

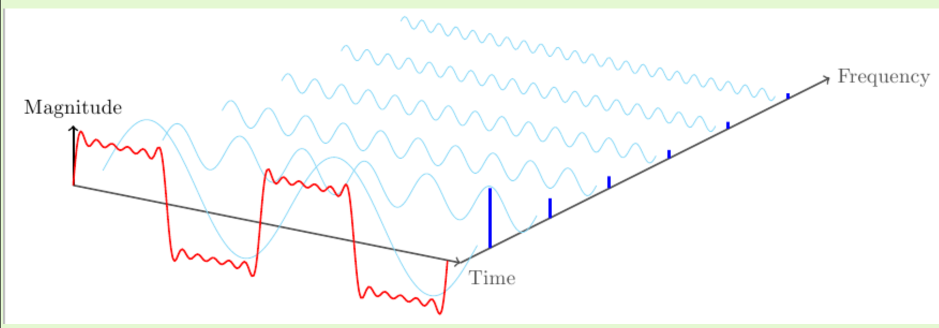
---

white 255

black 0

white

black 0



http://i.stack.imgur.com/Ofkna.png

PORTLAND STATE UNIVERSITY

DCTs

- DCT ← discrete cosine transform F

$f \; \boxed{x,y} \, 8 \times 8 \xrightarrow{DCT} \boxed{u,v}$ ← coefficients

$$F(u,v) = \frac{C_u}{2}\frac{C_v}{2}\sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y)\cos\left[\frac{(2x+1)u\pi}{16}\right]\cos\left[\frac{(2y+1)v\pi}{16}\right]$$

$$C_u = \begin{cases} \frac{1}{\sqrt{2}} & if\ u = 0 \\ 1 & if\ u > 0 \end{cases} \qquad Cv = \begin{cases} \frac{1}{\sqrt{2}} & if\ v = 0 \\ 1 & if\ v > 0 \end{cases}$$

- IDCT Inverse DCT

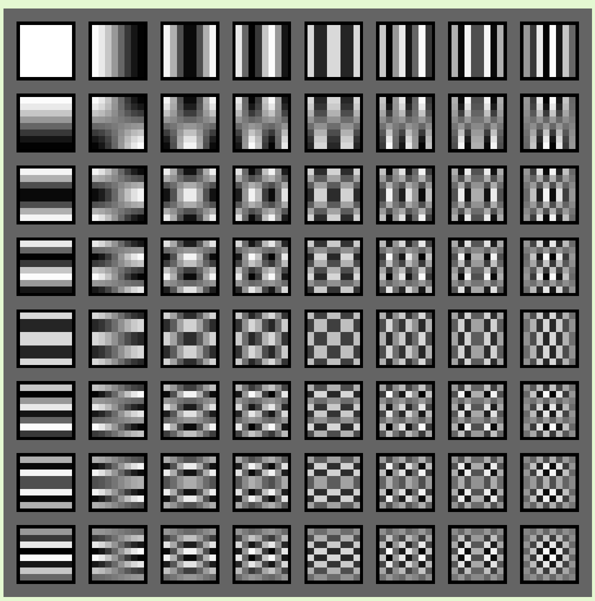
$$f(x,y) = \sum_{u=0}^{7}\sum_{v=0}^{7} \frac{C_u}{2}\frac{C_v}{2} F(u,v)\cos\left[\frac{(2x+1)u\pi}{16}\right]\cos\left[\frac{(2y+1)v\pi}{16}\right]$$

$$C_u = \begin{cases} \frac{1}{\sqrt{2}} & if\ u = 0 \\ 1 & if\ u > 0 \end{cases} \qquad Cv = \begin{cases} \frac{1}{\sqrt{2}} & if\ v = 0 \\ 1 & if\ v > 0 \end{cases}$$

PORTLAND STATE UNIVERSITY

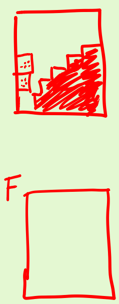
DCT Basis Functions

Any 8x8 spatial block can be made up of a summation of these patterns



spatial domain

f



F

PORTLAND STATE UNIVERSITY

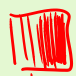
155	155	155	155	155	155	155	155
155	155	155	155	155	155	155	155
155	155	155	155	155	155	155	155
155	155	155	155	155	155	155	155
155	155	155	155	155	155	155	155
155	155	155	155	155	155	155	155
155	155	155	155	155	155	155	155
155	155	155	155	155	155	155	155

Non-zero coefficient of the DCT pattern in upper left

✓	0	0	0				
0	0	0	0				
0	0	0	.	.			
0	0			ALL	0!		
0	0						
0	0						
0	0						
0	0						

PORTLAND STATE UNIVERSITY

155	155	155	155	0	0	0	0
155	155	155	155	0	0	0	0
155	155	155	155	0	0	0	0
155	155	155	155	0	0	0	0
155	155	155	155	0	0	0	0
155	155	155	155	0	0	0	0
155	155	155	155	0	0	0	0
155	155	155	155	0	0	0	0



Non-zero

helping to give sharp edge

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
:							:
0	0	0	0	0	0	0	0

PORTLAND STATE UNIVERSITY

Quantization

spatial $\xrightarrow{\text{DCT}}$ frequency

[0, 255] → [-2047, 2048] floating pt. → ultimately want [0→255] values

8 × 8 → 8 × 8

Amount of DCT basis function pattern in

Quantization

Quantize coefficients COEFF[i,j]

user defined

$$COEFF[i,j] \,/\, (QM[i,j] \cdot QV)$$

COEFF

300	

QM

20	99	
95	80	
3		500
	400	1000

COEFF (i, j)

570	320	. . .
150	27	
⋮		

QM [i, j]

20	30	. . .
30	40	
⋮		

QV ← user define

"best" if QV = 1

coefficients to encode

28	10	
5	0	

$$COEFF(i,j) \,/\, (QM(i,j) \cdot QV)$$

QV = 4

7	2	
1	0	

Some real examples

155	155	155	155	155	155	155	155
155	155	155	155	155	155	155	155
155	155	155	155	155	155	155	155
155	155	155	155	155	155	155	155
155	155	155	155	155	155	155	155
155	155	155	155	155	155	155	155
155	155	155	155	155	155	155	155
155	155	155	155	155	155	155	155

1240	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

155	155	155	155	0	0	0	0
155	155	155	155	0	0	0	0
155	155	155	155	0	0	0	0
155	155	155	155	0	0	0	0
155	155	155	155	0	0	0	0
155	155	155	155	0	0	0	0
155	155	155	155	0	0	0	0
155	155	155	155	0	0	0	0

620	562	0	-197	0	132	0	-112
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

PORTLAND STATE UNIVERSITY

Quantization…

I have added some actual examples to the notes here…
The left matrix is the standard (real quantization matrix) I called this QM In class. The middle matrix would be COEFF that I called in class…

- the lossy part of JPEG / MPEG
- Take a quantization matrix and divide, each respective cell by the one in the quantization matrix.

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

620	562	0	-197	0	132	0	-112
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Pre-quantization

38	51	0	-13	0	3	0	-2
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Post-quantization

PORTLAND STATE UNIVERSITY

Quantization level

I have added some actual examples to the notes here…
This slide shows the difference between Q=1 and Q=4

- Multiply the quantization matrix by a larger number for each cell

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 = 1

620	562	0	-197	0	132	0	-112
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Pre-quantization

38	51	0	-13	0	3	0	-2
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Post-quantization

Q = 4

620	562	0	-197	0	132	0	-112
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

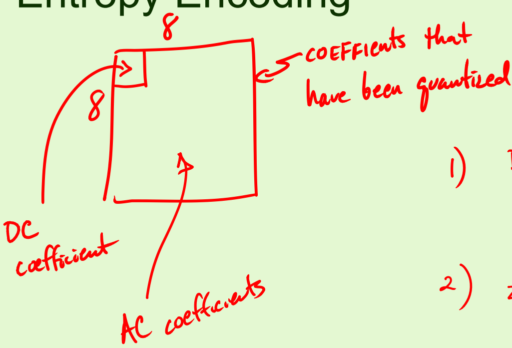
Pre-quantization

9	12	0	-4	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	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Post-quantization

Portland State University

Entropy Encoding

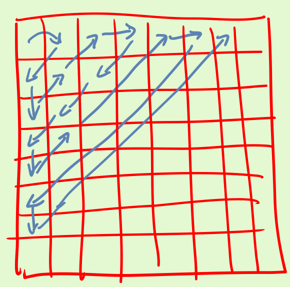


1) DC coefficient
differentially encode with respect to last block
Huffman encode difference

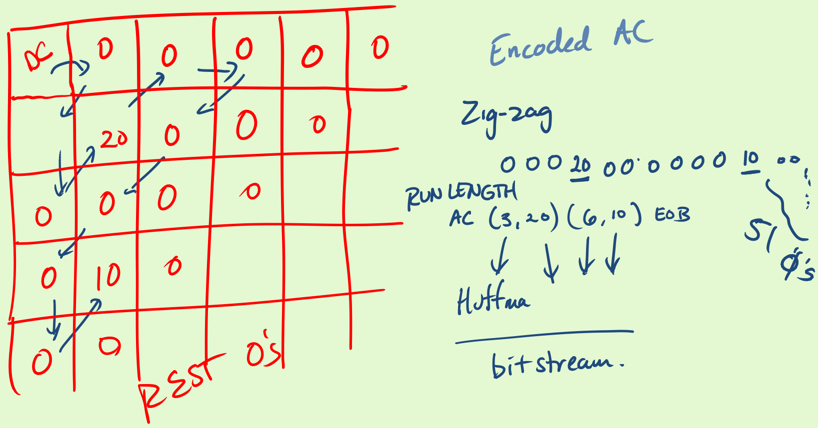
2) zig-zag reorder AC-coefficients

3) non-zero coefficient need to be encoded

RLE to non-zero coefficients
+
then Huffman encode
Run, + value



Portland State University



PORTLAND STATE UNIVERSITY

JPEG File

byte		
1-2	Start of Image (SOI) Marker	0xFFD8
3-4	APP0: App Segment	0xFFE0
5-6	Length	Length starting at this byte
7-11	Identifier	0x4A46494600 (null terminated “JFIF”)
12-13	Version	Typically 0x0101 (version 1.01)
14	Units	0: none; 1: inch; 2: cm
15-16	Xdensity	Horizontal pixel density
17-18	Ydensity	Vertical pixel density
19	Xthumbnail	Thumbnail size (x direction)
20	Ythumbnail	Thumbnail size (ydirection)
21-(20+3n)	Thumbnail	Thumbnail data (R, G, B, R, G, B, …) n = Xthumbnail * Ythumbnail

PORTLAND STATE UNIVERSITY

JPEG Data

Quantization Marker	0xFFDB
Quantization table used	Defines the quantization table
Huffman Table Marker	0xFFC4
Huffman table used	Defines the Huffman table used
Start of Frame Marker	0xFFC0
Rest of image data	Actual DCT, quantized, zig-zag reordered, and Huffman compressed data

PORTLAND STATE UNIVERSITY