JPEG IMAGE COMPRESSION

N. C. State University

CSC557 ♦ Multimedia Computing and Networking

Fall 2001

Lecture # 14

JPEG

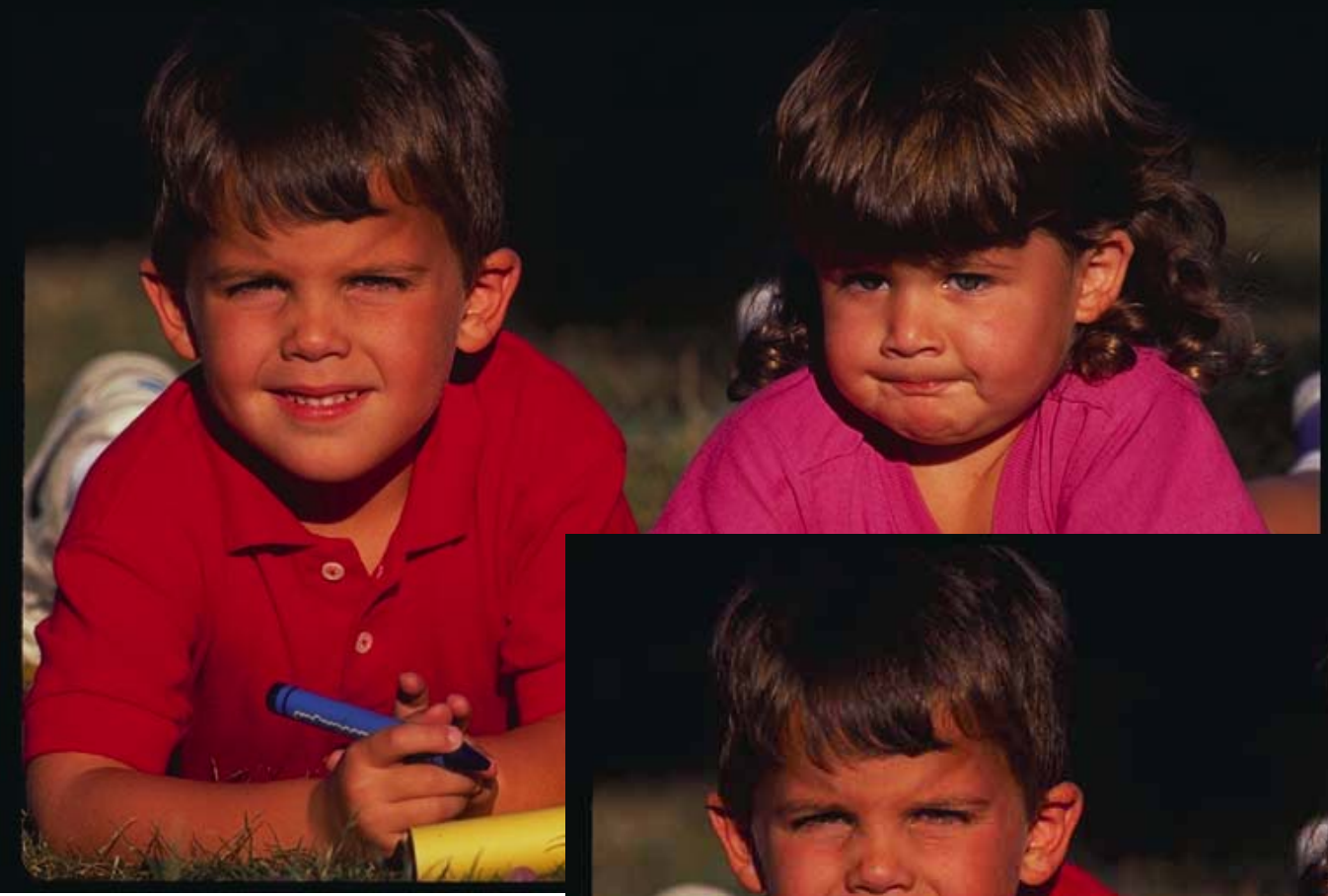
- First international digital image compression standard
- Typical compression: 10:1 to 50:1
- Lossy mode (usually)
  - but hardly noticeable
  - blockiness apparent at high compression ratios


1.1 MB
original
113 KB
JPEG-compressed

Original: 1.1MB

JPEG-compressed:
81KB

Original: 1.1MB



JPEG-compressed:
34 KB

JPEG Compression Modes

- Sequential (baseline) (only one we will look at)
- Progressive (successive refinement)
- Hierarchical (multiple image resolutions)
- Lossless

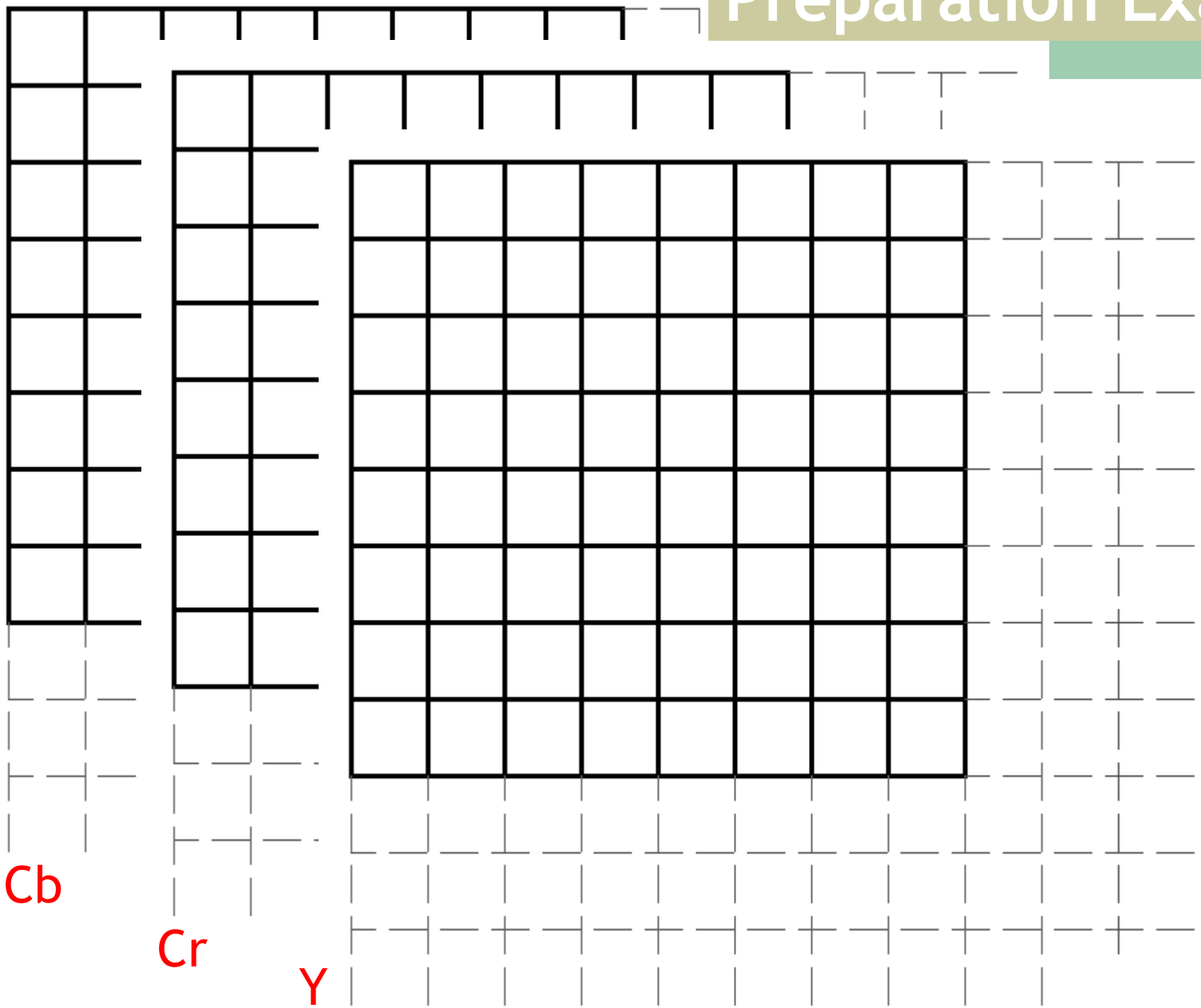
JPEG Steps

1. Image preparation (subsampling) (*lossy*)
2. Transform to frequency domain (DCT) (~lossless)
3. Quantization (*lossy*)
4. Zig-zag ordering and run-length encoding (lossless)
5. Huffman encoding (lossless)

Step 1. Image Preparation

- Color separation (convert RGB to YCbCr)
- Subsampling Cb and Cr (4:2:0 subsampling)
- Subtract 128 from each pixel value
  - shifts to -128 to +127 range
- Partition into 8x8 pixel blocks, which will now be processed independently

Preparation Example



Example Results

- An 8x8 block of pixel luminance values after image prep

48	53	58	62	64	64	64	64
53	60	62	65	68	65	65	65
59	64	69	72	67	65	65	65
68	70	71	69	69	68	68	68
68	69	70	71	71	64	64	64
70	70	70	70	69	66	66	66
71	71	70	72	71	66	66	66
71	71	70	70	72	67	67	67

(a)

Step 2. Discrete Cosine Transform

- Transform 8x8 image block into a frequency representation. Why?
  - human vision is most sensitive to middle frequencies
  - when compress, sacrifice precision of high frequencies
- Why DCT instead of DFT?
  - requires only real (rather than complex) arithmetic, faster to compute
  - has slightly better "energy compaction" → more of the information is packed into fewer of the components
- DCT basis functions
  - Cosines only
  - 8x8 = 64 basis functions

DCT Basis Functions (1-D)

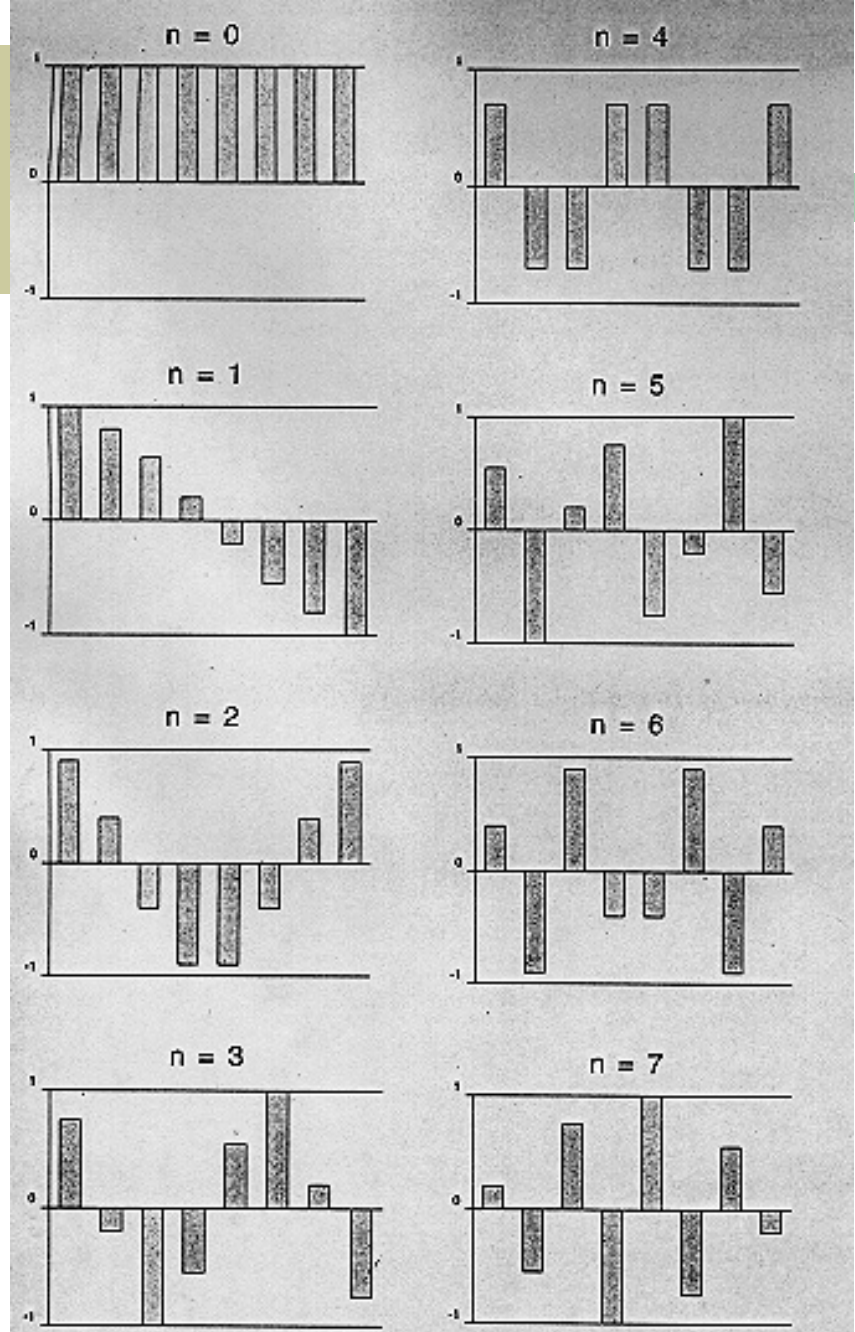
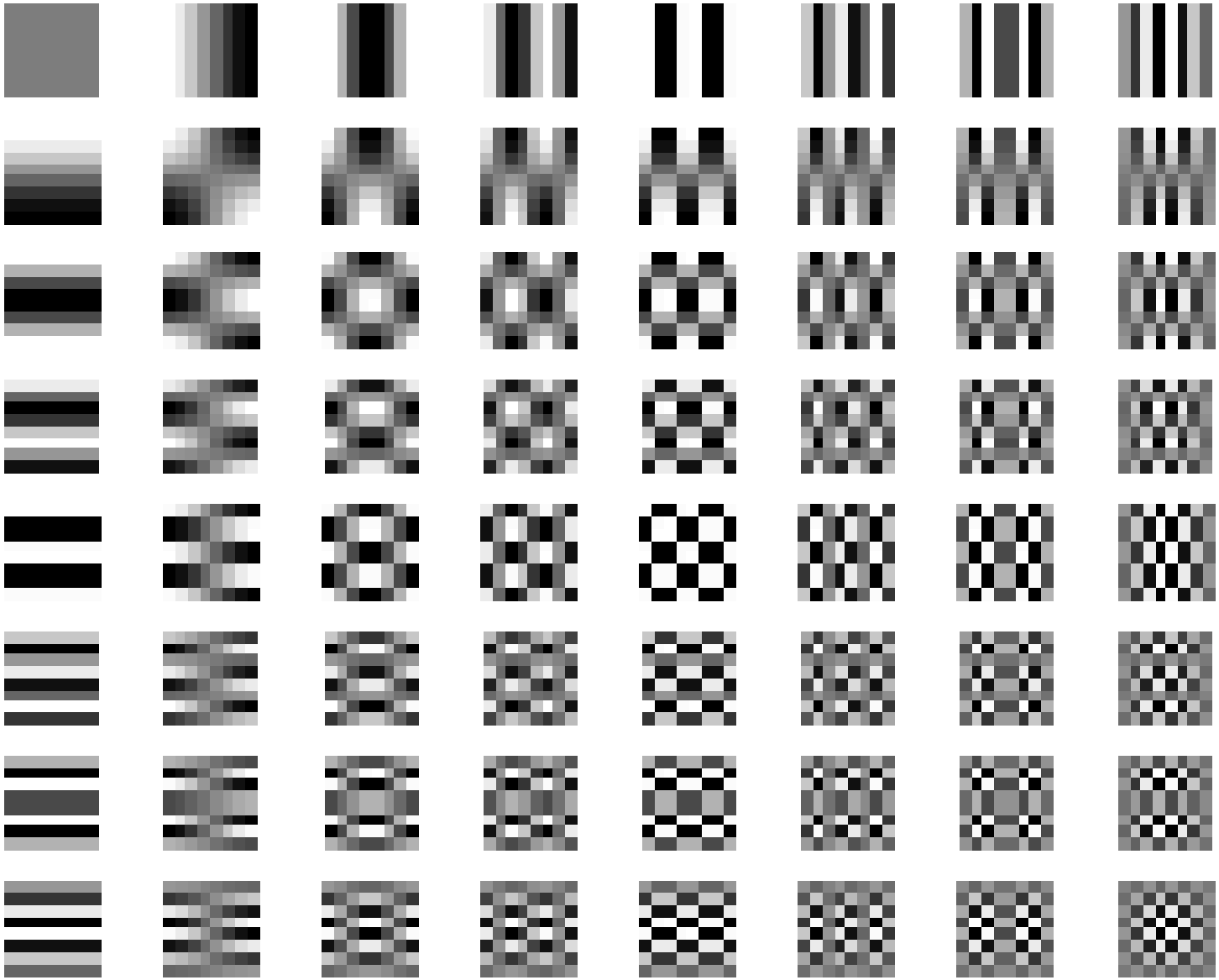


FIGURE 7.22 1-D cosine basis functions.

DCT Basis Functions (2-D)



DCT Equation for 8x8 Blocks

- Notation
  - p(x,y) = value of pixel in position (x,y)
  - H(u,v) = amplitude of the (u,v) basis function
- Method: correlate image with basis functions

$$H(u,v) = \frac{1}{4} C(u)C(v) * \sum_{x=0}^{7}\sum_{y=0}^{7} p(x,y) \times \left[\cos\left[\frac{(2x+1)u\pi}{16}\right] \cos\left[\frac{(2y+1)v\pi}{16}\right] \right]$$

$$C(a) = \sqrt{2}/2 \text{ if } a==0, \text{ else } C(a) = 1$$

DCT (cont.)

- H(0,0) (U.L.H. corner) is amplitude of lowest frequency component
  - Average intensity of whole 8x8 block
  - Also called the "DC coefficient"
- H(7,7) is amplitude of the highest frequency component

DCT (cont.)

- Results of DCT are in range -1024 to +1023

-492.4	-1.0	-12.0	-5.2	2.1	-1.7	-2.7	1.3
-22.6	-17.5	-6.2	-3.2	-2.9	-0.1	0.4	-1.2
-11.0	-9.3	-1.6	1.5	0.2	-0.9	-0.6	-0.1
-7.0	-1.9	0.2	1.5	0.9	-0.0	-0.0	0.3
-0.6	-0.8	1.5	1.6	-0.1	-0.7	0.6	1.3
1.8	-0.2	1.6	-0.3	-0.8	1.5	1.0	-1.0
-1.3	-0.4	-0.3	-1.5	-0.5	1.7	1.1	-0.8
-2.6	1.6	-3.8	-1.8	1.9	1.2	-0.6	-0.5

(b)

Some DCT Output Examples

On our website

Step 3. Quantization

- Quantization = integer division
  - divide DCT output H(u,v) by [t(u,v) / Q], with rounding
- User specifies the 8x8 quantization table T, and "quality factor" Q
  - T and Q not fixed by the standard
- Larger value for t(u,v) means lower quality, more compression for the u,vth basis component
- Larger Q means higher quality, less compression overall
- Larger values of T used for higher frequencies
  - clusters non-zero values in ULH corner

Example Quantization Table For Luminance

TABLE 9.6 Luminance quantization table.

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

Quantization Example

-492.4	-1.0	-12.0	-5.2	2.1	-1.7	-2.7	1.3
-22.6	-17.5	-6.2	-3.2	-2.9	-0.1	0.4	-1.2
-11.0	-9.3	-1.6	1.5	0.2	-0.9	-0.6	-0.1
-7.0	-1.9	0.2	1.5	0.9	-0.0	-0.0	0.3
-0.6	-0.8	1.5	1.6	-0.1	-0.7	0.6	1.3
1.8	-0.2	1.6	-0.3	-0.8	1.5	1.0	-1.0
-1.3	-0.4	-0.3	-1.5	-0.5	1.7	1.1	-0.8
-2.6	1.6	-3.8	-1.8	1.9	1.2	-0.6	-0.5

(b)

TABLE 9.6 Luminance quantization table.

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

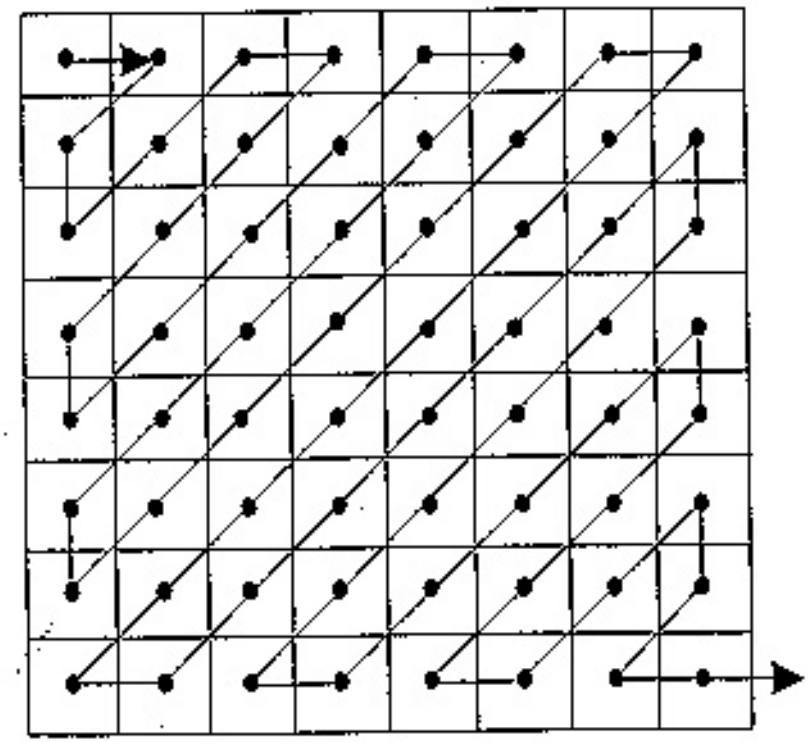
=

-31	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	0	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

(c)

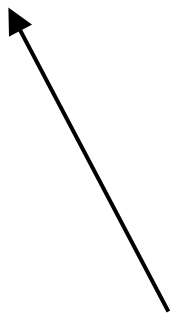
Step 4. Run-length Encoding

- Zig-zag ordering groups zeros together
- DC coefficient is differentially encoded



RLE (cont.)

- Encode runs of up to 16 consecutive zeros
  - 16 zeros ==> "run not finished yet"
- EOB code ==> "rest of the block is zeros"
- <+3> <1,-2> <0,-1> <0,-1> <0,-1> <2,-1> <0,-1> eob



"Old" value = -34, difference = -31 - (-34) = +3

Step 5. Huffman Encoding

- RLE results are huffman (variable-length) encoded
  - (Runlength,non-zero value "size") replaced by a code
  - Non-zero value follows this code
- (Example: see Crane, Tables 9.8-13)

JPEG Decoding

- Reverse the order of the encoding steps

1. Huffman decode (lossless, need table)
2. RLE decode (lossless), un-zig-zag order
3. Dequantization (multiply by T/Q, need both)
4. Inverse DCT (lossless)
5. Un-sub-sample Chroma information, "glue" blocks together, color conversion if necessary

Inverse DCT

$$p(x,y) = \frac{1}{4}\sum_{u=0}^{7}\sum_{v=0}^{7} C(u)C(v)\ H(u,v)\cos[\frac{(2x+1)u\pi}{16}]\cos[\frac{(2y+1)v\pi}{16}]$$

C(a) = $\sqrt{2}/2$ if a==0, else C(a) = 1

Decoding Example

After inverse quantization

−496	0	−10	0	0	0	0	0
−24	−12	0	0	0	0	0	0
−14	−13	0	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

(a)

After inverse DCT

50	52	55	58	60	61	62	62
57	58	61	63	64	65	64	64
65	66	67	69	69	68	67	66
70	70	71	71	70	68	66	65
70	70	70	70	69	66	64	63
68	69	69	69	68	66	64	62
68	68	69	70	69	68	66	65
68	69	71	72	72	71	69	68

(b)

Losses Caused During Compression

−2	1	3	4	4	3	2	2
−4	2	1	2	4	0	1	1
−6	−2	2	3	−2	−3	−2	−1
−2	0	0	−2	−1	0	2	3
−2	−1	0	1	2	−2	0	1
2	1	1	1	1	0	2	4
3	3	1	2	2	−2	0	1
3	2	−1	−2	0	−4	−2	−1

(c)

Sources Of Info

- Recommended
  - [Crane97] *A Simplified Approach to Image Processing*
    - Chapter 7 (DCT)
    - Chapter 9 (JPEG)
- Optional
  - [Kou95] *Digital Image Compression*
  - [Pennebaker93] *JPEG Still Image Data Compression Standard*