Text and Images

CSC 790

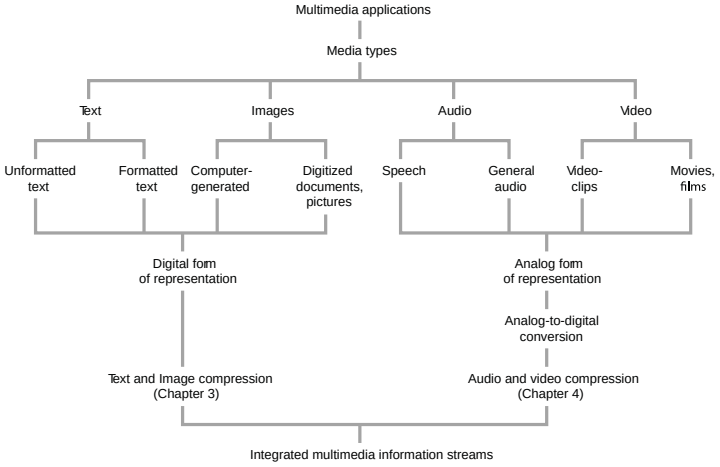
WAKE FOREST UNIVERSITY

Department of Computer Science

Fall 2009

Multimedia

- Multimedia is information using one or more of the following
  - Text, images, audio, video
- These types can be further classified into the following



Media Representations and Conversions

- For each media type we will consider the following
  - Analog form, digital form (encode/decode), and compression

  What does each mean?

  In what order do they occur?

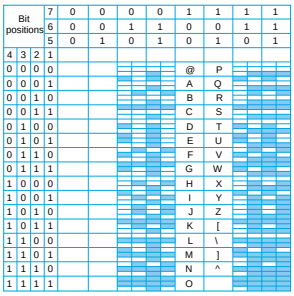
Text

- There are three general types of text used
  - Unformatted, formatted, and hypertext

    What is the analog form of text?

- Digital form of unformatted text can be based on tables

Bit positions	7	0	0	0	0	1	1	1	1
	6	0	0	1	1	0	0	1	1
	5	0	1	0	1	0	1	0	1
4	3	2	1							
0	0	0	0	NUL	DLE	SP	0	@	P	\	p
0	0	0	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	STX	DC2		2	B	R	b	r
0	0	1	1	ETX	DC3	#	3	C	S	c	s
0	1	0	0	EOT	DC4	$	4	D	T	d	t
0	1	0	1	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	ACK	SYN	&	6	F	V	f	v
0	1	1	1	BEL	ETB		7	G	W	g	w
1	0	0	0	BS	CAN	(	8	H	X	h	x
1	0	0	1	HT	EM	)	9	I	Y	i	y
1	0	1	0	LF	SUB	*	:	J	Z	j	z
1	0	1	1	VT	ESC	+	;	K	[	k	{
1	1	0	0	FF	FS	,	<	L	\	l	|
1	1	0	1	CR	GS	—	=	M	]	m	}
1	1	1	0	SO	RS	.	>	N	^	n	~
1	1	1	1	SI	US	/	?	O		o	DEL



Bit positions	7	0	0	0	0	1	1	1	1
	6	0	0	1	1	0	0	1	1
	5	0	1	0	1	0	1	0	1
4	3	2	1						
0	0	0	0				@	P			
0	0	0	1				A	Q			
0	0	1	0				B	R			
0	0	1	1				C	S			
0	1	0	0				D	T			
0	1	0	1				E	U			
0	1	1	0				F	V			
0	1	1	1				G	W			
1	0	0	0				H	X			
1	0	0	1				I	Y			
1	0	1	0				J	Z			
1	0	1	1				K	[			
1	1	0	0				L	\			
1	1	0	1				M	]			
1	1	1	0				N	^			
1	1	1	1				O				

  - Where every symbol has a fixed length **codeword**

- *But this depends on the encoding method...*

Encoding

- Assume codewords could have variable length
  - *Most frequently used symbols should have short codewords*
    Example encoding technique that follows this principle?
- Shannon developed an equation to measure encoding **entropy**, H

$$H = -\sum_{i=1}^{n} p_i \log_2 p_i$$

  - Where n is the number of different symbols in the data stream and p_i is the probability of symbol i occurring

Assume the symbols $\{a, b, c, d, e, f\}$ have the probability $\{0.25, 0.25, 0.125, 0.125, 0.125, 0.125\}$ respectively. Given the encoding is $\{$`10, 11, 010, 011, 000, 001`$\}$ then what is the average number of bits per codeword and entropy?

Huffman Encoding

- Minimizes the average number of bits/codeword, 1954
  - Variable length codewords
  - However, no codeword is a prefix of another
    What?
- Build a binary tree, assign symbols based on probabilities

Create two ascending ordered queues, q_1 and q_2
Create a node for each symbol (leaf nodes)
Add all the nodes to q_1
while length(q_1) > 1 **AND** length(q_1) > 1
    Remove two lowest nodes, s_1 and s_2
    Create a new node (combine s_1 and s_2) and add to q_2
    Make the combined node the parent of s_1 and s_2
Remaining node is the root of the tree

Huffman Example

What is the Huffman encoding for the symbols $\{a, b, c, d, e f\}$ *with probabilities* $\{0.25, 0.25, 0.125, 0.125, 0.125, 0.125\}$

What is the average number of bits per codeword? entropy?

General Compression Techniques

- Given network bandwidth is limited, compression is necessary
  - Reduce the amount of information required to represent data

    Is Huffman a compression technique?

- Two general forms of compression
  - Lossless, reversible so no data is lost
  - Lossy, normally cannot reproduce original (true copy)

  What is the advantage of each? When would you use either?

Entropy Encoding

- Lossless and has two forms, run-length and statistical
- Statistical encoding
  - Use variable length codewords based on probability (Huffman)
- Run-length encoding
  - A symbol many occur more often than others
  - If the same symbol occurs consecutively, considered a **run**
  - Instead of repeating the symbol, transmit symbol and count

original	$aaaaabbaaaaaabbbbbbbbbabbbab$
run-length	$(5,a),(2,b),(6,a),(9,b),(1,a),(3,b),(1,a),(1,b)$

Can this be used with Huffman encoding?

Another Run-Length

- Assume you are transmitting black (1) or white (0) pixels
  - Assume white pixels have a higher occurrence
- We could code the runs of white pixels are follows

Run	Length	Codeword
0	0	00
01	1	01
001	2	10
0001	3	11

- Assumes the maximum run-length is 3

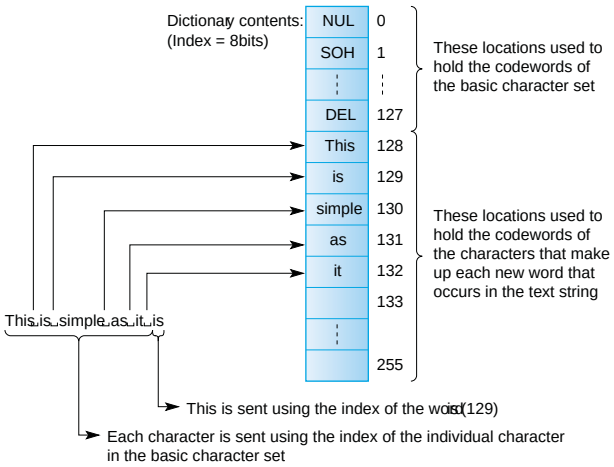
Of course the the above runs could repeat, so we could...

Lempel-Ziv Encoding

- Run-length encoding based on characters or strings
  - Table created containing all possible strings
  - Assign codewords to strings, thus it is *dictionary based*

  Problems?

- Lempel-Ziv-Welsh encoding builds the dictionary dynamically
  - Dictionary initial contains all possible characters
  - Assume the text to encode starts with "`This is as it is`"
  - Encode `'T'`, `'h'`, `'i'`, `'s'` using alphabet codewords
  - Encode `' '`, since space is not a alphanumeric character, add string `'This'` to the dictionary



  - If the string `'This'` occurs again, use the codeword 128

- Encoder and decoder follow the same algorithm

  What is the problem with this technique?

Another LZW Example

- Assume only three characters A, B, and C
  - Dictionary has initial size of 2 bits

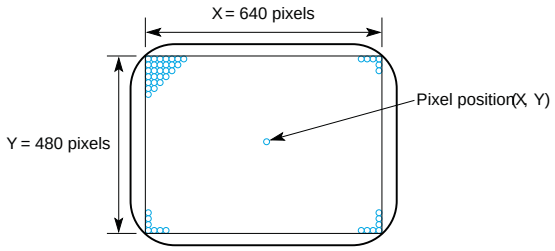
00	A
01	B
10	C

- Encode the sequence is AABABCA

Current Sequence	Next Char	Output	Comments
' '	A		
A	A	00	add AA to dictionary as 11
A	B	00	add AB to dictionary as 100
B	A	01	add BA to dictionary as 101
AB	C	100	add ABC to dictionary 110
C	A	010	add CA to dictionary 111
A	' '	000	

Images

- When in digital form, represented as an array of pixels
  - Each pixel can represent a color
- More bits per pixel (depth) allows more colors

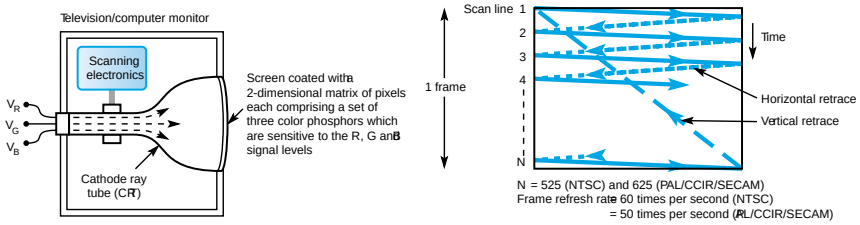


X × Y = spatial resolution of screen
8 bits per pixel = 256 different colors

Standard	Resolution	Number of Colors	Memory per image
VGA	$640 \times 480 \times 8$	256	307.2 kB
XGA	$640 \times 480 \times 16$	64 k	614.4.2 kB
	$1024 \times 768 \times 8$	256	786.432 kB
SVGA	$640 \times 480 \times 16$	64 k	960 kB
	$1024 \times 768 \times 8$	256	786.432 kB
	$1024 \times 768 \times 24$	16 M	2359.269 kB

Image Display

- Eye sees one color if the set of red, green, and blue are mixed
  - Therefore to display, must project the three colors

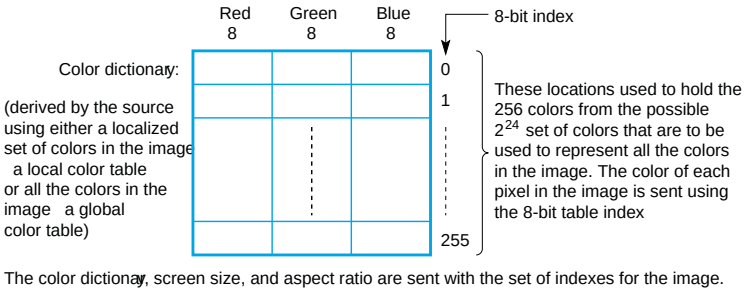


- Each pixel on the display has a *phosphor triad*
  - Beam (raster) energizes the triad (raster-scan)

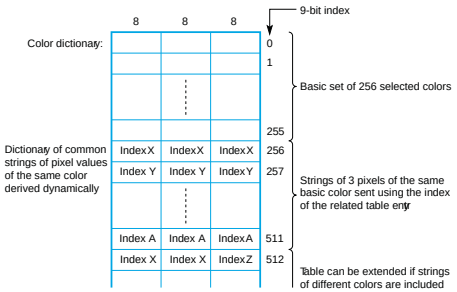
What is the connection between a pixel and the three colors?

Graphics Interchange Format

- Graphic Interchange Format (GIF) is widely used for the web
- Color is supported using only 8 bits
  - GIF reduces the colors palette to 256 most used colors
  - 8 bits used to index the color table



- Can also use LZW compression for *strings of colours*

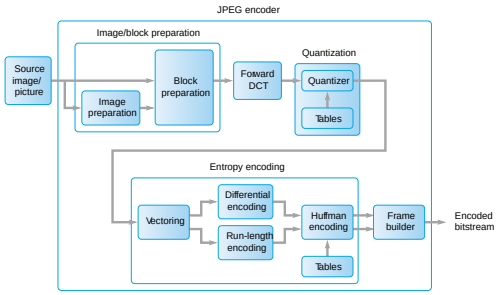


- Which started patent issues

  Is GIF a lossy or lossless encoding method?

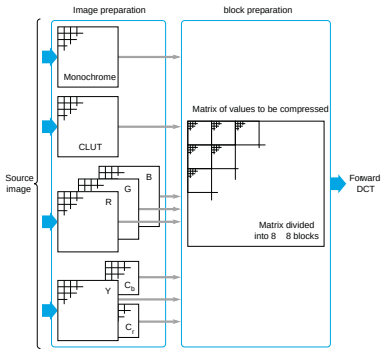
Joint Photographic Experts Group

- Joint Photographic Experts Group (JPEG) is a lossy representation
  - However a lossless version does exist
- There are five main stages
  - Image/block preparation, forward DCT, quantization, entropy encoding, and frame building



JPEG Image Preparation

- An image consists of an 2D array of pixels
  - 1 array for monochrome, 3 arrays for color
- The image is then divided into 8×8 blocks



Why 8×8?

- Blocks set sequentially to the DCT

Fourier Analysis

In the 19th Century, Jean-Baptiste Fourier proved that any reasonably behaved periodic function $g(t)$, with period T, can be constructed by summing a number of sines and cosines.

$$g(t) = \frac{1}{2}c + \sum_{n=1}^{\infty} a_n \sin(2\pi n f t) + \sum_{n=1}^{\infty} b_n \cos(2\pi n f t) \quad (1)$$

Where $f = \frac{1}{T}$ is the fundamental frequency and, a_n and b_n are the sine and cosine amplitudes of the nth terms (harmonics).

$$c = \frac{2}{T} \int_0^T g(t)dt \quad (2)$$

$$a_n = \frac{2}{T} \int_0^T g(t) \sin(2\pi n f t)dt \quad (3)$$

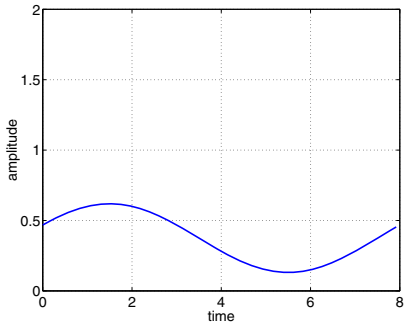
$$b_n = \frac{2}{T} \int_0^T g(t) \cos(2\pi n f t)dt \quad (4)$$

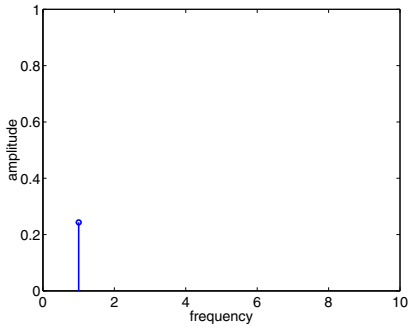
The Fourier analysis of the signal [01100010] yields the coefficients

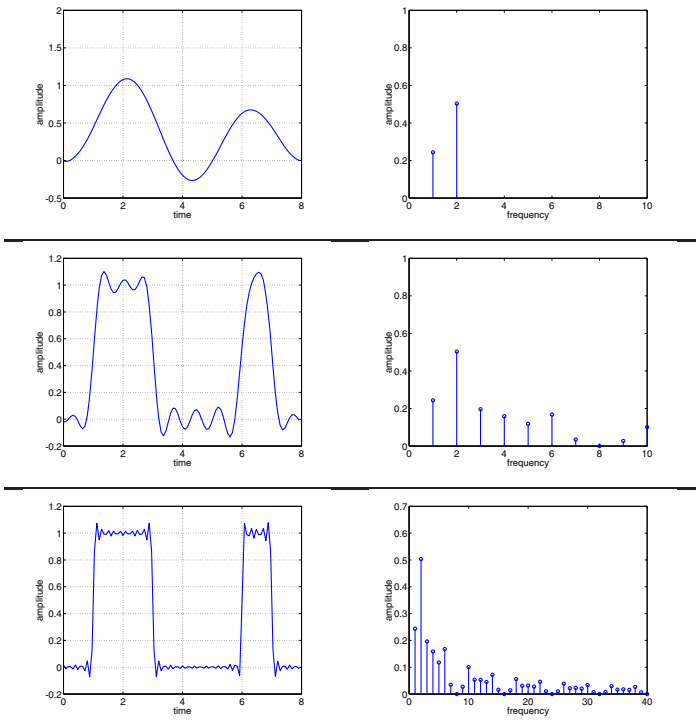
$$c = \frac{3}{4}$$

$$a_n = \frac{1}{\pi n}\left[\cos\left(\frac{\pi n}{4}\right) - \cos\left(\frac{3\pi n}{4}\right) + \cos\left(\frac{6\pi n}{4}\right) - \cos\left(\frac{7\pi n}{4}\right)\right]$$

$$b_n = \frac{1}{\pi n}\left[\sin\left(\frac{3\pi n}{4}\right) - \sin\left(\frac{\pi n}{4}\right) + \sin\left(\frac{7\pi n}{4}\right) - \sin\left(\frac{6\pi n}{4}\right)\right]$$







Tranforms

- Previous slides show how a single wave can be transformed
  - Complex wave can be created from multiple *simple* waves
  - Each wave had a different amplitude, frequency, and phase
  - This easily applies to sound, but also images
  - For images, the *wave* can represent changes in color intensity
- A transform is needed, Fourier, Karhunen-Löve, or DCT are all possible *but which one...*
  - DCT is used for JPEG because it is easy to implement and provides reasonable *compression* performance

    Is a transform a form of compression?

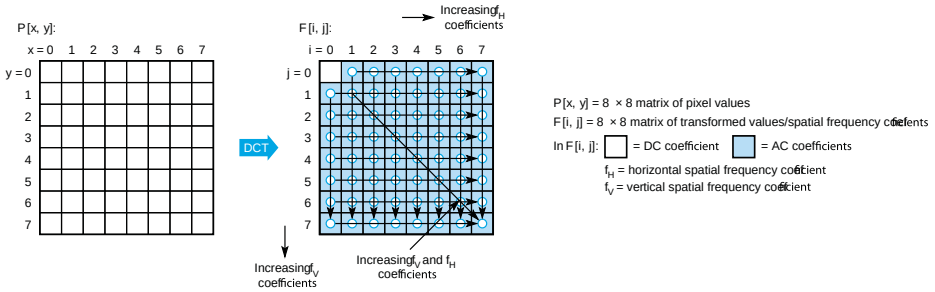
Discrete Cosine Transform

- Perform a Discrete Cosine Transform (DCT) on each block
  - Let $f_{i,j}$ be the DCT of the $p_{i,j}$ pixel

$$f_{i,j} = \frac{1}{4} c_i c_j \sum_{x=0}^{7} \sum_{y=0}^{7} p_{x,y} \cos \frac{(2x+1)i\pi}{16} \cos \frac{(2y+1)j\pi}{16}$$

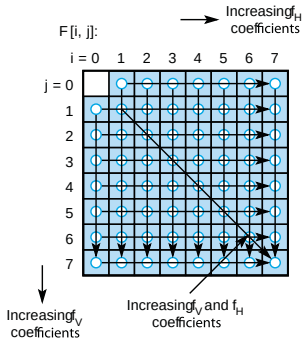
$$c_i, c_j = \begin{cases} \frac{1}{\sqrt{2}} & i, j = 0 \\ 1 & \text{otherwise} \end{cases}$$

- The transform done for each element in the array



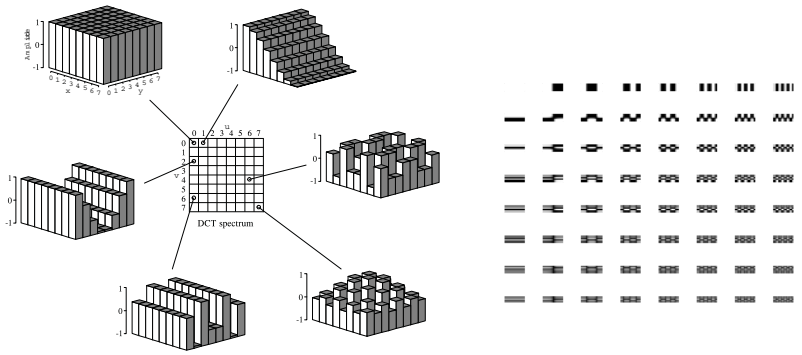
A Few Things to Consider

- For $i = j = 0$ the cosine terms are zero
  - The value at $f_{0,0}$ is the summation of all the values
  - All other f values have a coefficient, called AC coefficients
- Only horizontal frequencies coefficients present for $i = 1, ..., 7$
- Only vertical frequencies coefficients present for $j = 1, ..., 7$

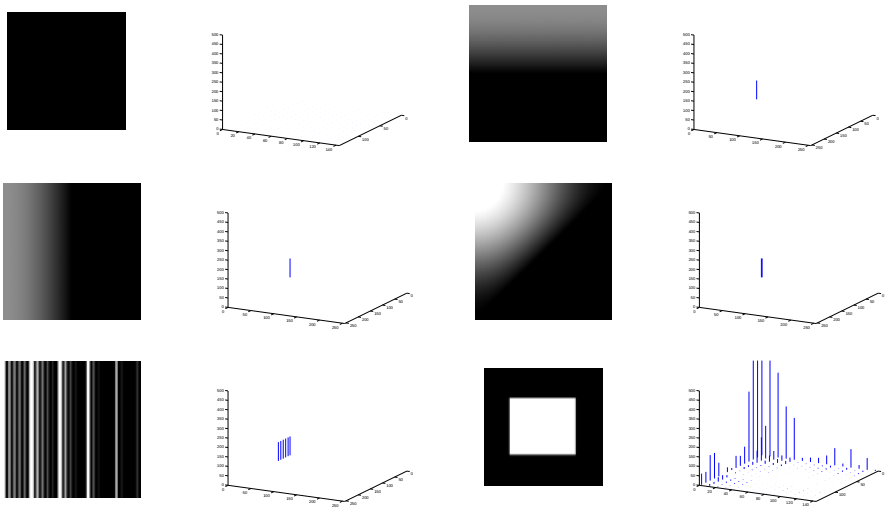


DCT Basis Function

- The DCT values are real numbers, called the **basis function**
  - Each $f(i, j)$ value is the amplitude for a wave
- Consider the following $f(i.j)$ values
  - $f(0, 0)$ is the DC component
  - $f(1, 0)$ is $\frac{1}{2}$ cycle of a cosine wave in one direction
  - $f(0, 1)$ is the same but rotated by 90 degrees



Example Image and DCT



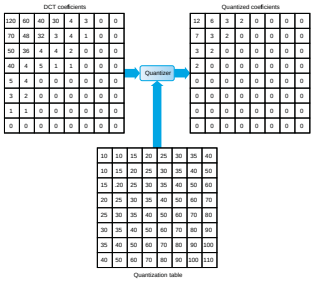
Is the DCT a form of compression?

Quantization

- The DCT values are then quantized
  - Quantization done based on perception of human eye

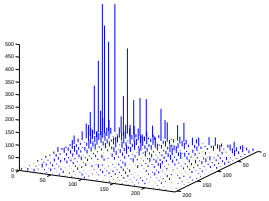
    Similar to what compression technique?

- Frequency coefficients need to be above threshold for the eye
  - Frequencies below these thresholds can be dropped
  - Quantization table has the threshold values

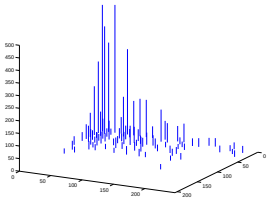


Dead Zone

- A **dead-zone quantizer** sets frequency to zero if below threshold
  - For example, the original picture and DCT

  - All frequency coefficients below 20 set to zero

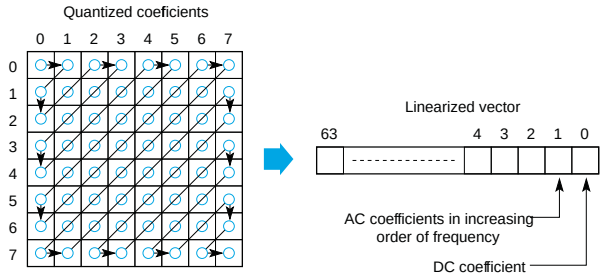
How does quantization save space?

DCT Encoding

- DCT values ($f_{i,j}$) are encoded to remove redundancy
  - DC components are differentially encoded

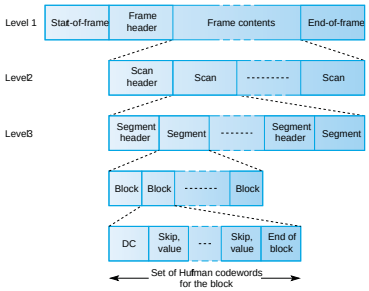
    What?

  - AC components are Huffman (run-length) encoded
- DCT values are processed using a **zig-zag** scan



JPEG Frame

- Frame builder takes JPEG information and creates a frame



- Hierarchical structure containing headers for each layer
  - Frame header contains the width and height
  - Second level header contains the quantization table
  - Multiple 8×8 blocks grouped to form a segment at level 3