Multimedia communications
ECP 610

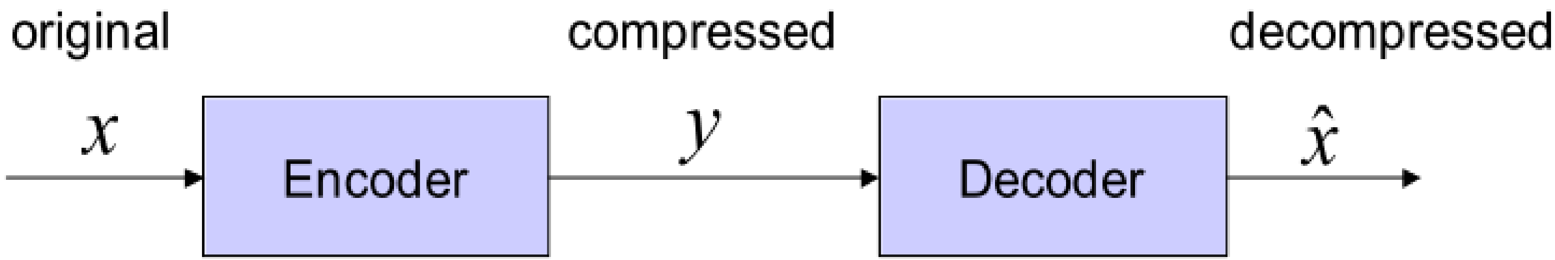
Omar A. Nasr

omaranasr@ieee.org

Feb, 2015

Lossy compression-Huffman coding

Definition of compression



- Lossless compression $x = \hat{x}$
  - Also called entropy coding, reversible coding.
- Lossy compression $x \neq \hat{x}$
  - Also called irreversible coding.
- Compression ratio = $|x|/|y|$
  - $|x|$ is number of bits in x.

What is “good compression”?

- High compression ratio
- Complexity of the encoder
- Complexity of the decoder
- Speed of encoding
- Speed of decoding
- Quality

Lossless coding

- Basic definitions:
  - Binary code: each symbol is encoded as a binary string
  - Codeword: a binary string representing a certain symbol
  - Fixed length coding: each symbol is encoded using the same number of bits. All codewords have the same length
  - Variable length coding: different symbols uses codewords with different lengths
  - Uniquely decodable code: codewords can be decoded in only one way

Lossless coding

codebook

codewords

	a	b	c	d	e	f
Freq in '000s	45	13	12	16	9	5
a fixed-length	000	001	010	011	100	101
a variable-length	0	101	100	111	1101	1100

codewords

- Fixed length coding = (45+13+12+16+9+5)*3 = 300kbits
- Var length coding = 45*1+(13+12+16)*3+(9+5)*4 = 224kbits

Encoding and decoding

- Encoding: replacing the symbols by their codewords
- Decoding: getting the symbols back

$$
\begin{aligned}
C_1 &= \{a = 00,\ b = 01,\ c = 10,\ d = 11\}. \\
C_2 &= \{a = 0,\ b = 110,\ c = 10,\ d = 111\}. \\
C_3 &= \{a = 1,\ b = 110,\ c = 10,\ d = 111\}
\end{aligned}
$$

- Decode *010011* using C1
- Decode *1101111* using C3

Prefix codes

- Prefix codes: if no codeword is prefix to the other ☺

$$
\begin{aligned}
C_1 &= \{a = 00,\ b = 01,\ c = 10,\ d = 11\}. \\
C_2 &= \{a = 0,\ b = 110,\ c = 10,\ d = 111\}. \\
C_3 &= \{a = 1,\ b = 110,\ c = 10,\ d = 111\}
\end{aligned}
$$

- Is C2 a prefix code?
- Is it instantaneously decodable?
- Any prefix code is uniquely decodable

Huffman coding

- An "entropy coding" technique
- Assign just the correct number of bits to different symbols depending on their frequency

Character	***a***	***b***	***c***	***d***	***average***
Frequency	60	5	30	5	
Fixed length	00	01	10	11	2 bits
Variable length	0	110	10	1110	1.55 bits
Variable length 2	0	110	10	111	1.5 bits

- Huffman coding finds the minimum length prefix code in a systematic way!

Huffman coding

- Steps:

1- Pick two letters x, y from alphabet A with the smallest frequencies and create a subtree that has these two characters as leaves. Label the root of this subtree as z.

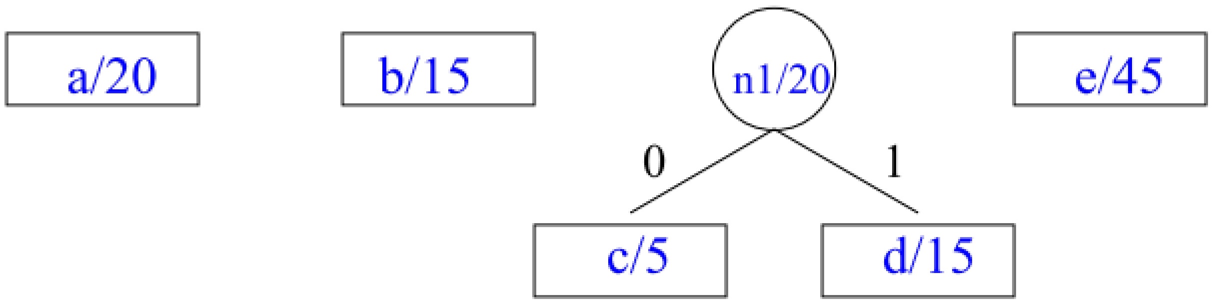
2- Set frequency $f(z) = f(x) + f(y)$

3- remove x,y and replace them with z with its new frequency

4- repeat (1) until the end

Example

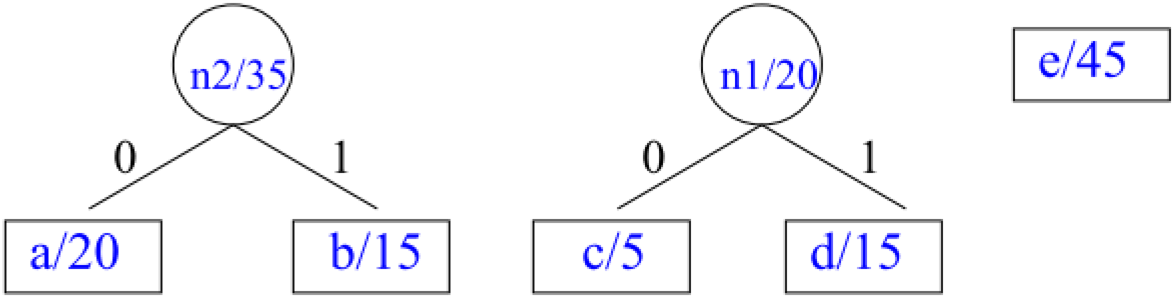
Let $A = \{a/20, b/15, c/5, d/15, e/45\}$



$$A_1 = \{a/20, b/15, n1/20, e/45\}$$

Example (cont.)

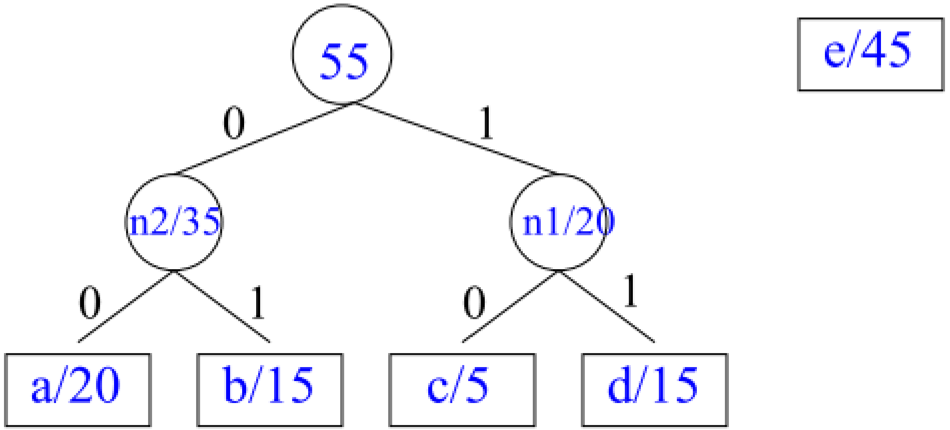
$$A_1 = \{a/20, b/15, n1/20, e/45\}$$



$$A_2 = \{n2/35, n1/20, e/45\}$$

Example (cont.)

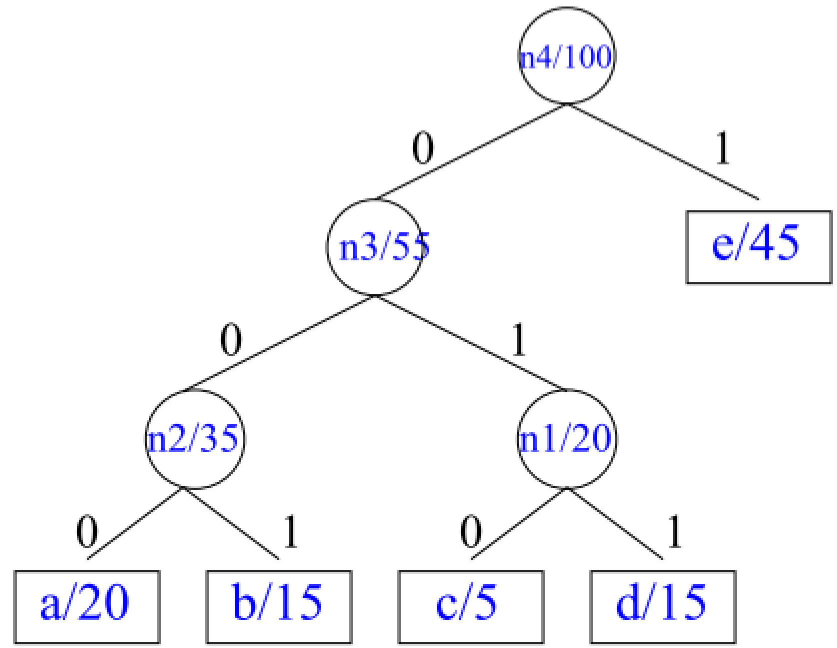
$$A_2 = \{n2/35, n1/20, e/45\}$$



$$A_3 = \{n3/55, e/45\}$$

$A_3 = \{n3/55, e/45\}$

Example (cont.)



$a = 000, b = 001, c = 010, d = 011, e = 1$

- Calculate the average length
- What do you think the most suitable prob distributions for Huffman coding?

Variations of Huffman coding

- Huffman coding of vectors:
  - Instead of considering individual symbols, consider two or more symbols together!
  - For example, in our case, the alphabet will be

"aa, ab, ac, ad, ba, …., dd"

  - The average length in the example will be (2.04 bits)
- Conditional Huffman coding
  - For *N* symbols, build *N* different Huffman codebooks based on the different conditional probabilities
- Adaptive Huffman coding

Discrete Cosine Transform

Discrete Fourier Transform

$$X(k) \triangleq \sum_{n=0}^{N-1} x(n)e^{-j2\pi nk/N}, \quad k = 0, 1, 2, \ldots, N-1$$

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)e^{j2\pi nk/N}, \quad n = 0, 1, 2, \ldots, N-1$$

DCT (1D)

- Discrete cosine transform

$$C(u) = \alpha(u) \sum_{x=0}^{N-1} f(x) \cos\left(\frac{(2x+1)u\pi}{2N}\right)$$

$$\alpha(k) = \begin{cases} \sqrt{\frac{1}{N}} & \text{if } k = 0 \\ \sqrt{\frac{2}{N}} & \text{otherwise} \end{cases}.$$

- The strength of the 'u' sinusoid is given by C(u)
  - Project f onto the basis function
  - All samples of f contribute the coefficient
  - C(0) is the zero-frequency component – the average value!

DCT (1D)

- Consider a digital image such that one row has the following samples

Index	0	1	2	3	4	5	6	7
Value	20	12	18	56	83	10	104	114

- There are 8 samples so N=8
- u is in [0, N-1] or [0, 7]
- Must compute 8 DCT coefficients: C(0), C(1), …, C(7)
- Start with C(0)

$$C(0) = \sqrt{\frac{1}{N}} \sum_{x}^{N-1} f(x)$$

DCT (1D)

$$
\begin{aligned}
C(0) &= \sqrt{\tfrac{1}{8}} \sum_{x=0}^{7} f(x) \cos\left(\frac{(2x+1)\cdot 0\pi}{2\cdot 8}\right) \\
&= \sqrt{\tfrac{1}{8}} \sum_{x=0}^{7} f(x) \cos(0) \\
&= \sqrt{\tfrac{1}{8}} \cdot \{f(0) + f(1) + f(2) + f(3) + f(4) + f(5) + f(6) + f(7)\} \\
&= .35 \cdot \{20 + 12 + 18 + 56 + 83 + 110 + 104 + 115\} \\
&= 183.14
\end{aligned}
$$

DCT (1D)

- Repeating the computation for all u we obtain the following coefficients

Spatial domain

f(0)	f(1)	f(2)	f(3)	f(4)	f(5)	f(6)	f(7)
20	12	18	56	83	110	104	114

Frequency domain

$C(0)$	$C(1)$	$C(2)$	$C(3)$	$C(4)$	$C(5)$	$C(6)$	$C(7)$
183.1	-113.0	-4.2	22.1	10.6	-1.5	4.8	-8.7

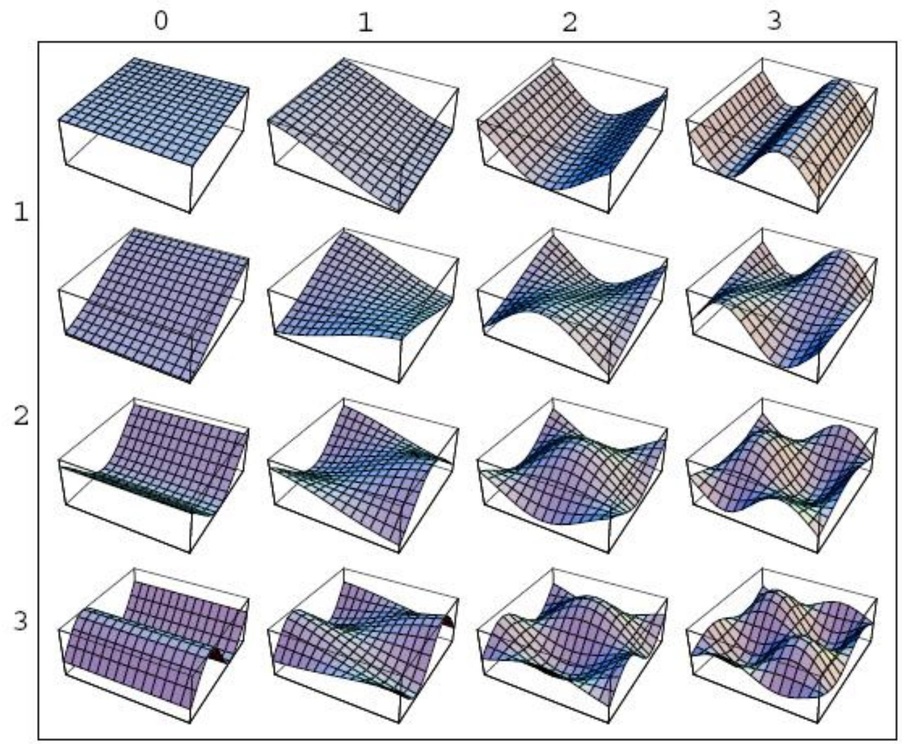
DCT (2D)

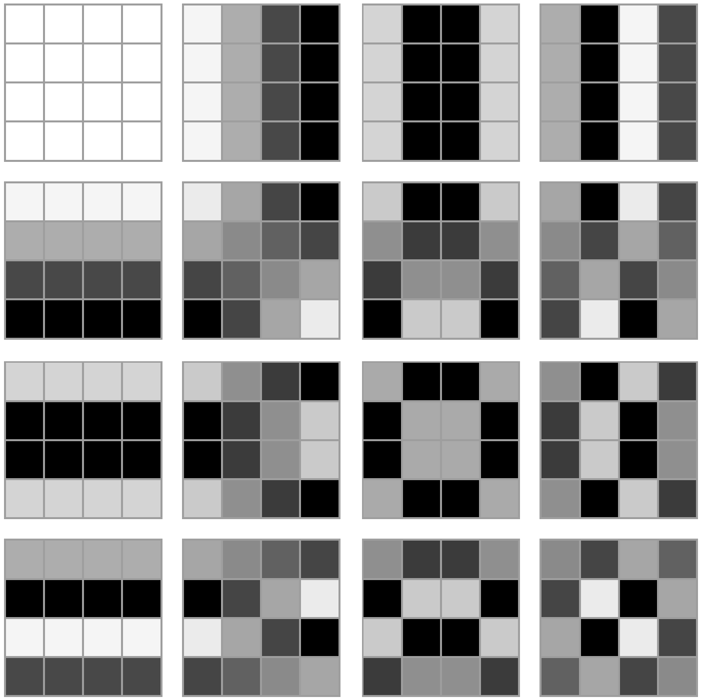
- The 2D DCT is given below where the definition for alpha is the same as before

$$C(u,v) = \alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left(\frac{(2x+1)u\pi}{2N}\right)\cos\left(\frac{(2y+1)v\pi}{2N}\right)$$

- For an MxN image there are MxN coefficients
- Each image sample contributes to each coefficient
- Each (u,v) pair corresponds to a ‘pattern’ or ‘basis function’

DCT basis functions (patterns)



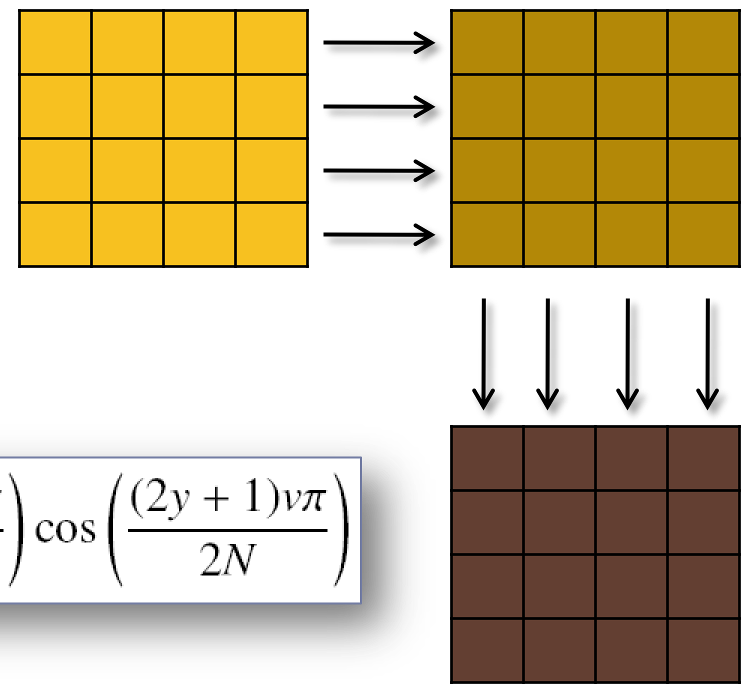


Basis functions

Basis patterns (imaged functions)

Separability

- The DCT is separable
  - The coefficients can be obtained by computing the 1D coefficients for each row
  - Using the row-coefficients to compute the coefficients of each column (using the 1D forward transform)

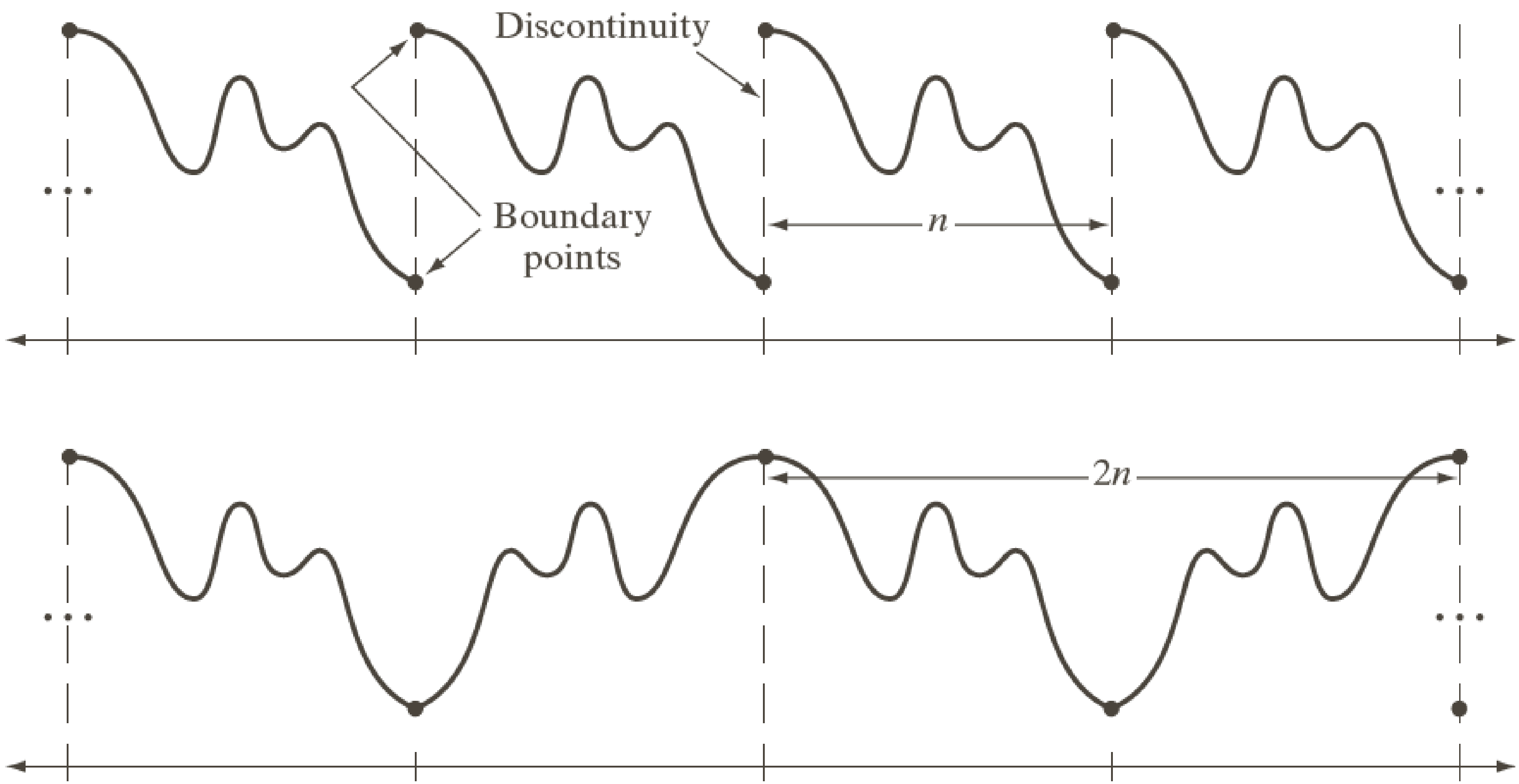


$$C(u,v) = \alpha(u)\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \cos\left(\frac{(2x+1)u\pi}{2N}\right) \cos\left(\frac{(2y+1)v\pi}{2N}\right)$$

Invertability

- The DCT is invertible
  - Spatial samples can be recovered from the DCT coefficients

$$
\begin{aligned}
f(x) &= \sum_{u=0}^{N-1} \alpha(u)C(u)\cos\left(\frac{(2x+1)u\pi}{2N}\right) \\
f(x,y) &= \sum_{u=0}^{N-1}\sum_{v}^{N-1} \alpha(u)\alpha(v)C(u,v)\cos\left(\frac{(2x+1)u\pi}{2N}\right)\cos\left(\frac{(2y+1)v\pi}{2N}\right)
\end{aligned}
$$


Discontinuity
Boundary
points
n
2n

Summary of DCT

- The DCT provides energy compaction
  - Low frequency coefficients have larger magnitude (typically)
  - High frequency coefficients have smaller magnitude (typically)
  - Most information is compacted into the lower frequency coefficients (those coefficients at the 'upper-left')

- Compaction can be leveraged for compression
  - Use the DCT coefficients to store image data but discard a certain percentage of the high-frequency coefficients!
  - JPEG does this