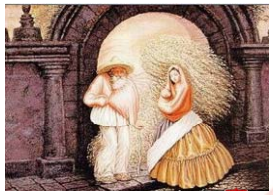
NANYANG TECHNOLOGICAL UNIVERSITY

Applied Machine Vision

Session 4



presented by

Xie Ming

Associate Professor
School of Mechanical & Aerospace Engineering

Image Compression

- Part 1: Data Compression
- Part 2: Image Compression
- Part 3: Video Compression

Part 1:

Data Compression

Learning Objectives

- Data
- Compression
- LZW Algorithm
- Huffman Algorithm
- Q & A

Data

What is digital data?

- Your answer:

a)

What is digital data?

- Digital data refer to strings of symbols, which may represent texts, drawings, graphics, and images.

How to represent digital data?

- Your answer:

a)

A	
B	
C	
D	
E	
…	
0	
1	
2	
…	
8	
9	
…	

How to represent digital data?

- A common solution is to use a dictionary of codes such as ASCII codes.

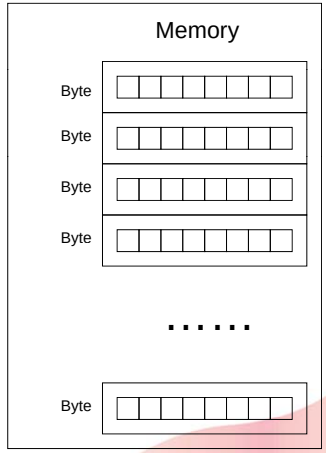
Char	ASCII Code
A	65
B	66
C	67
D	68
E	69
…	
0	48
1	49
2	50
…	
8	56
9	57
…	

How to store digital data?

- Your answer:

  a)

  b)

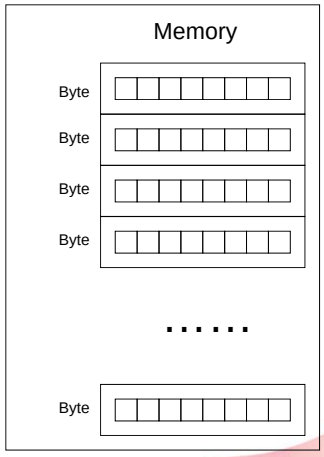


How to store digital data?

- In general, we can say

  a) a digital data is stored as a string of bytes.

  b) a digital data is stored as a string of bits.

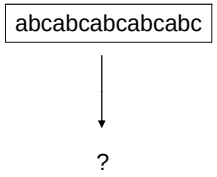


Compression

What is data compression?

- Your answer:

  a)



What is data compression?

- By definition, data compression is to transform data from the original representation into another form of representation with reduced size.

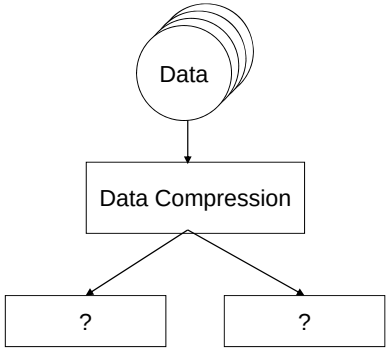
abcabcabcabcabc = 5 x abc

Why is there a need to compress data?

- Your answer:

  a)

  b)

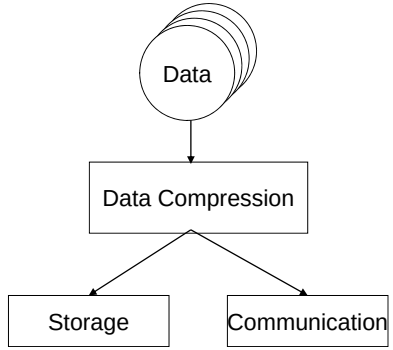


Why is there a need to compress data?

- In general, data compression can be motivated by two reasons:

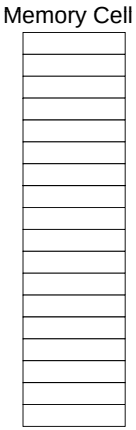
  a) data communication

  b) data storage

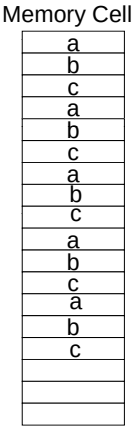


- Example 1:

How to store the string of characters “abcabcabcabcabc”?



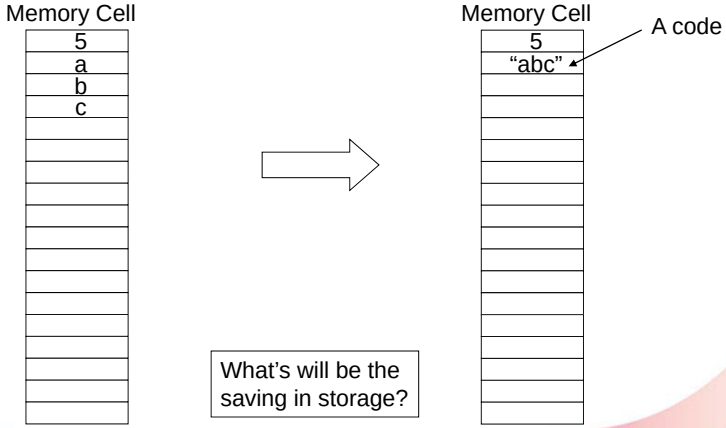
Solution A:



Can we reduce the usage of memory cells?

Solution B:

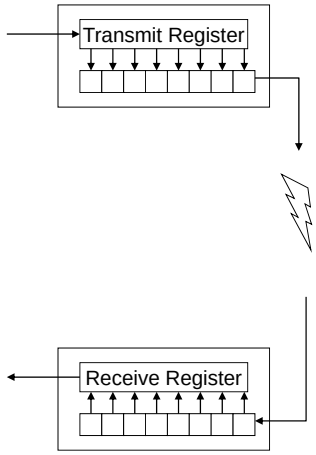
abcabcabcabcabc = 5 x abc



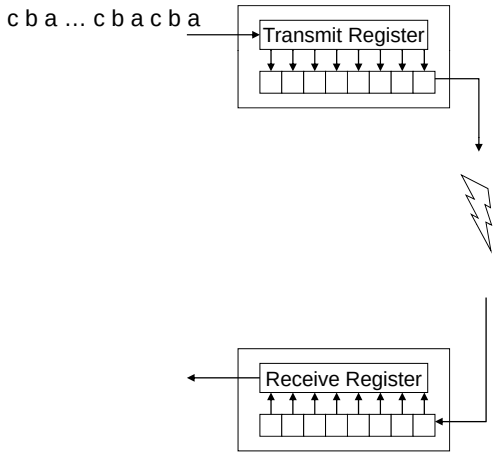
What’s will be the saving in storage?

- Example 2:

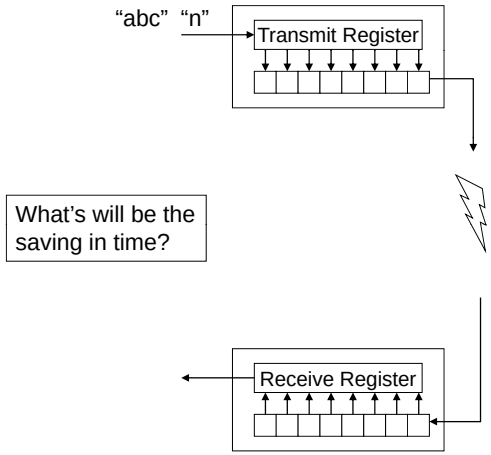
How to serially transmit the string of characters “abcabcabc … abc” from one unit to another unit over long distance?



Solution A:



Solution B:



What's will be the saving in time?

LZW Algorithm

What is the necessary condition?

- Your answer:

  a)

  b)

To compress "abcdefac"

To compress "abababab"

What is the necessary condition?

- Obviously, the necessary condition is that

  a) there are symbols which repeat inside a data, and

  b) each repetition should be more then once.

To compress “abcdefac”
To compress “abababab”

(A symbol is any combination of characters)

What is the key issue?

- Your answer:

  a)

To compress “abcdefac”
To compress “abababab”

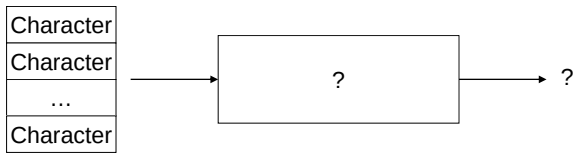
What is the key issue?

- The key issue is to identify symbols, which repeat inside a digital data.

To compress “abcdefac”
To compress “abababab”

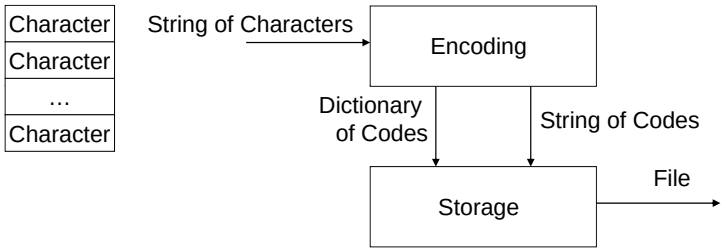
What is the procedure of data compression?

- Your answer:



What is the procedure of data compression?

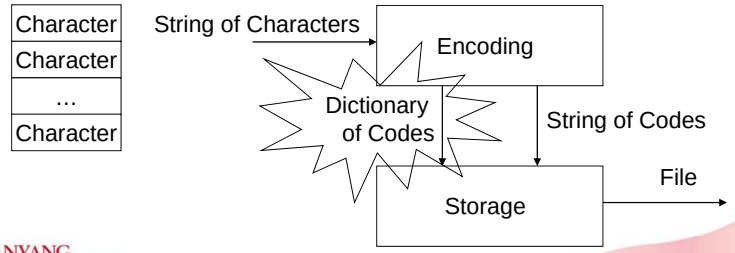
- Answer:



What is the dictionary of codes?

- Your answer:

?



What is the dictionary of codes?

- Answer:

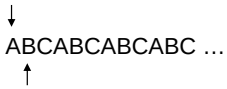
By default, the dictionary contains the ASCII codes.

During the encoding process, a new code will be created when a new symbol is encountered.

Dictionary

Symbol	code
Symbol	code
…	…
Symbol	code

- Example:

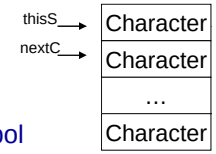


Step 1: Read “A”
Step 2: Read “B” (to create a new code for “AB”)
Step 3: Read “C” (to create a new code for “BC”)
Step 5: Read “A” (to create a new code for “CA”)
Step 6: Read “B”
Step 7: Read “C” (to create a new code for “ABC”)
…

How to create the dictionary of codes during compression?

- Answer:

  a) The dictionary initially contains the ASCII codes.



  b) Then, the encoding process uses two pointers:
  thisS: which contains the current symbol
  nextC: which points to the next character

  c) If “thisS+nextC” is new, to create a new code to the dictionary.

  d) If “thisS+nextC” exists, thisS = thisS+nextC.

How to create the dictionary of codes during compression?

- Example:

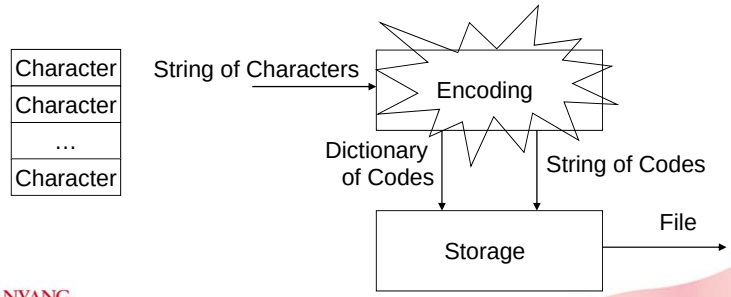
  ABCABCABCABC …

Step 1: Read “A” -> thisS
Step 2: Read “B” -> nextC
thisS+nextC = AB, which is new.
to create a new code for “AB” and
thisS = “B”
Step 3: Read “C” -> nextC
thisS+nextC = BC, which is new.
to create a new code for “BC” and
thisS = “C”
Step 5: Read “A” -> nextC
thisS+nextC = CA, which is new.
to create a new code for “CA”, and
thisS = “A”
Step 6: Read “B”-> nextC
this S+nextC = AB, which is not new
thisS = AB
Step 7: Read “C” -> nextC
thisS+nextC = ABC, which is new
to create a new code for “ABC”, and
thisS = “C”
…

How to encode a string of characters?

- Your answer:

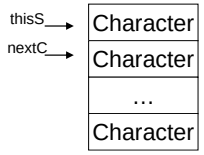
  ?



How to encode a string of characters?

- Answer:

  a) The dictionary initially contains the ASCII codes.



  b) Then, the encoding process uses two pointers:
  thisS: which contains the current symbol
  nextC: which points to the next character

  c) If “thisS+nextC” is new, to create a new code. And, do:
  - output the code of thisS
  - thisS = nextC

  d) If “thisS+nextC” is not new, thisS = thisS+nextC.

How to encode a string of characters?

- Example:

ABCABCABCABC …

```
Step 1: Read “A” -> thisS
Step 2: Read “B” -> nextC
        thisS+nextC = AB, which is new.
        to create a new code for “AB” and
        output Code(thisS) and do thisS = “B”
Step 3: Read “C” -> nextC
        thisS+nextC = BC, which is new.
        to create a new code for “BC” and
        output Code(thisS) and do thisS = “C”
Step 5: Read “A” -> nextC
        thisS+nextC = CA, which is new.
        to create a new code for “CA”, and
        output Code(thisS) and do thisS = “A”
Step 6: Read “B”-> nextC
        thisS+nextC = AB, which is not new
        thisS = AB
Step 7: Read “C” -> nextC
        thisS+nextC = ABC, which is new
        to create a new code for “ABC”, and
        output Code(thisS) and do thisS = “C”
…
```


How to encode a string of characters?

- Pseudo-program:

```
thisS = NIL;

while ( read a character to nextC )
{
        if “thisS+nextC” exists in the dictionary,
                thisS = “thisS+nextC”;
        else
                add “thisS+nextC” to the dictionary;
                output the code of thisS;
                thisS = nextC;
}
```


How to encode a string of characters?

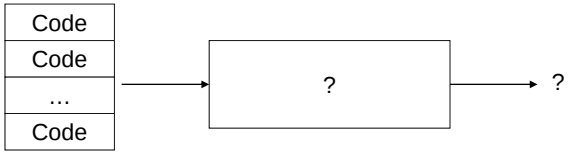
- Example:

Input string is “^WED^WE^WEE^WEB^WET”.

thisS	thisC	output	Code	symbol
NIL	^			
^	W	^	256	^W
W	E	W	257	WE
E	D	E	258	ED
D	^	D	259	D^
^	W			
^W	E	256	260	^WE
E	^	E	261	E^
^	W			
^W	E			
^WE	E	260	262	^WEE
E	^			
E^	W	261	263	E^W
W	E			
WE	B	257	264	WEB
B	^	B	265	B^
^	W			
^W	E			
^WE	T	260	266	^WET
T	EOF	T		

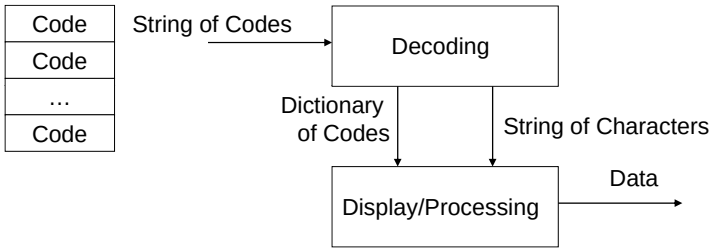
What is the procedure of data decompression?

- Your answer:



What is the procedure of data decompression?

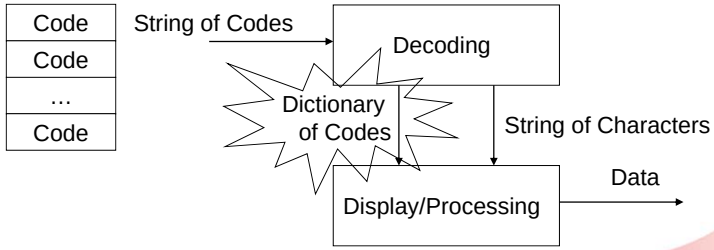
- Answer:



How to create the dictionary of codes during decompression?

- Your answer:

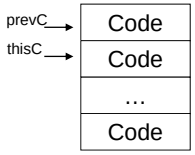
?



How to create the dictionary of codes during decompression?

- Answer:

a) The dictionary initially contains the ASCII codes.

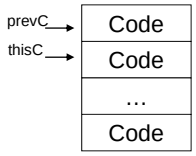


b) Then, the decoding process uses two pointers:
prevC: which points to the current code
thisC: which points to the next code

c) Add “Symbol(prevC)+first character of Sybmol(thisC)” to the dictionary and create a new code in the dictionary.

How to create the dictionary of codes during decompression?

- Pseudo-program:




```
string prevS, thisS;
int    prevC, thisC;

…

prevC = read in a code ;
prevS = Symbol(prevC) ;

while (there is still code to read)
{
        thisC = read in a code;
        thisS = Symbol(thisC);

        add “prevS+thisS[0]” to dictionary;
        prevS = thisS ;
}
```

How to create the dictionary of codes during decompression?

- Example:

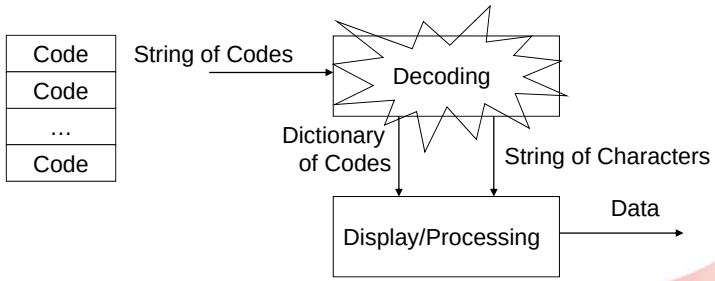
Input codes is "^WED<256>E<260><261><257>B<260>T".

prevC	thisC	Code	symbol
^	W	256	^W
W	E	257	WE
E	D	258	ED
D	<256>	259	D^
<256>	E	260	^WE
E	<260>	261	E^
<260>	<261>	262	^WEE
<261>	<257>	263	E^W
<257>	B	264	WEB
B	<260>	265	B^
<260>	T	266	^WET

How to decode a string of codes into a string of characters?

- Your answer:

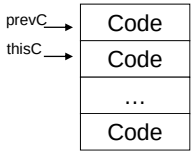
?



How to decode a string of codes into a string of characters?

- Answer:

a) The dictionary initially contains the ASCII codes.



b) Then, the decoding process uses two pointers:
prevC: which points to the current code
thisC: which points to the next code

c) Add "Symbol(prevC)+first character of Sybmol(thisC)" to the dictionary and create a new code in the dictionary.

d) Output Symbol(thisC) (except for the first code).

How to decode a string of codes into a string of characters?

- Pseudo-program:

```
string prevS, thisS;
int    prevC, thisC;

…

prevC = read in a code ;
prevS = Symbol(prevC) ;
output prevS;
while (there is still code to read)
{
        thisC = read in a code;
        thisS = Symbol(thisC);
        output thisS ;
        add "prevS+thisS[0]" to dictionary;
        prevS = thisS ;
}
```

How to decode a string of codes into a string of characters?

- Example:

Input codes is "^WED<256>E<260><261><257>B<260>T".

prevC	thisC	output	Code	symbol
	^	^		
^	W	W	256	^W
W	E	E	257	WE
E	D	D	258	ED
D	<256>	^W	259	D^
<256>	E	E	260	^WE
E	<260>	^WE	261	E^
<260>	<261>	E^	262	^WEE
<261>	<257>	WE	263	E^W
<257>	B	B	264	WEB
B	<260>	^WE	265	B^
<260>	T	T	266	^WET

What is the drawback of LZW algorithm?

- Your answer:

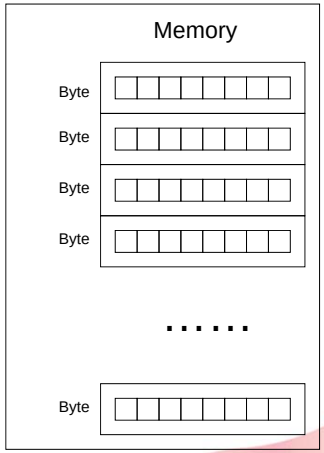
a)

ASCII Code:
- 8 bits
- fixed bit-length

a … b … d …… g …. k ….

What is the drawback of LZW algorithm?

- LZW algorithm is effective in reducing the number of symbols, which represent a data as a combination of bytes.
- LZW algorithm depends on the codes of fixed bit-length (e.g. ASCII).
- LZW algorithm is not effective in reducing the number of bits, which represent the same data as a combination of bits.



Huffman Algorithm

What is a ASCII code?

- Your answer:

  a)

32	00100000	20h	space
33	00100001	21h	!
34	00100010	22h	"
35	00100011	23h	#
36	00100100	24h	$
37	00100101	25h	%
38	00100110	26h	&
39	00100111	27h	'
40	00101000	28h	(
41	00101001	29h	)
42	00101010	2Ah	*
43	00101011	2Bh	+
44	00101100	2Ch	,
45	00101101	2Dh	-
46	00101110	2Eh	.
47	00101111	2Fh	/
48	00110000	30h	0
49	00110001	31h	1
50	00110010	32h	2
51	00110011	33h	3
52	00110100	34h	4
53	00110101	35h	5
54	00110110	36h	6
55	00110111	37h	7
56	00111000	38h	8
57	00111001	39h	9
58	00111010	3Ah	:
59	00111011	3Bh	;
60	00111100	3Ch	<
61	00111101	3Dh	=
62	00111110	3Eh	>
63	00111111	3Fh	?

What is a ASCII code?

- A ASCII code refers to a value of fixed bit-length (i.e. 7 bits), which represents an alpha-numeric character in English.

32	00100000	20h	space
33	00100001	21h	!
34	00100010	22h	"
35	00100011	23h	#
36	00100100	24h	$
37	00100101	25h	%
38	00100110	26h	&
39	00100111	27h	'
40	00101000	28h	(
41	00101001	29h	)
42	00101010	2Ah	*
43	00101011	2Bh	+
44	00101100	2Ch	,
45	00101101	2Dh	-
46	00101110	2Eh	.
47	00101111	2Fh	/
48	00110000	30h	0
49	00110001	31h	1
50	00110010	32h	2
51	00110011	33h	3
52	00110100	34h	4
53	00110101	35h	5
54	00110110	36h	6
55	00110111	37h	7
56	00111000	38h	8
57	00111001	39h	9
58	00111010	3Ah	:
59	00111011	3Bh	;
60	00111100	3Ch	<
61	00111101	3Dh	=
62	00111110	3Eh	>
63	00111111	3Fh	?

What is a ASCII code?

- Example 1:

65	01000001	41h	A
66	01000010	42h	B
67	01000011	43h	C
68	01000100	44h	D
69	01000101	45h	E
70	01000110	46h	F
71	01000111	47h	G
72	01001000	48h	H
73	01001001	49h	I
74	01001010	4Ah	J
75	01001011	4Bh	K
76	01001100	4Ch	L
77	01001101	4Dh	M
78	01001110	4Eh	N
79	01001111	4Fh	O
80	01010000	50h	P
81	01010001	51h	Q
82	01010010	52h	R
83	01010011	53h	S
84	01010100	54h	T
85	01010101	55h	U
86	01010110	56h	V
87	01010111	57h	W
88	01011000	58h	X
89	01011001	59h	Y
90	01011010	5Ah	Z

What is a ASCII code?

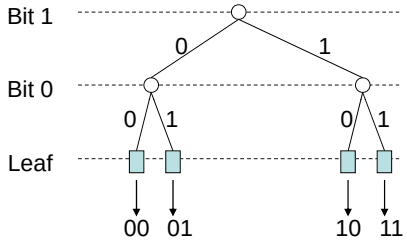
- Example 2:

97	01100001	61h	a
98	01100010	62h	b
99	01100011	63h	c
100	01100100	64h	d
101	01100101	65h	e
102	01100110	66h	f
103	01100111	67h	g
104	01101000	68h	h
105	01101001	69h	i
106	01101010	6Ah	j
107	01101011	6Bh	k
108	01101100	6Ch	l
109	01101101	6Dh	m
110	01101110	6Eh	n
111	01101111	6Fh	o
112	01110000	70h	p
113	01110001	71h	q
114	01110010	72h	r
115	01110011	73h	s
116	01110100	74h	t
117	01110101	75h	u
118	01110110	76h	v
119	01110111	77h	w
120	01111000	78h	x
121	01111001	79h	y
122	01111010	7Ah	z
123	01111011	7Bh	{
124	01111100	7Ch	|
125	01111101	7Dh	}
126	01111110	7Eh	~
127	01111111	7Fh	

How to construct the tree representation of ASCII codes?

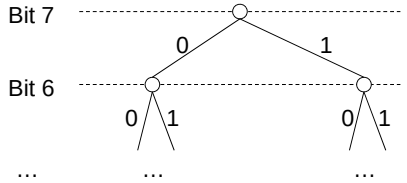
- Your answer:

  ?

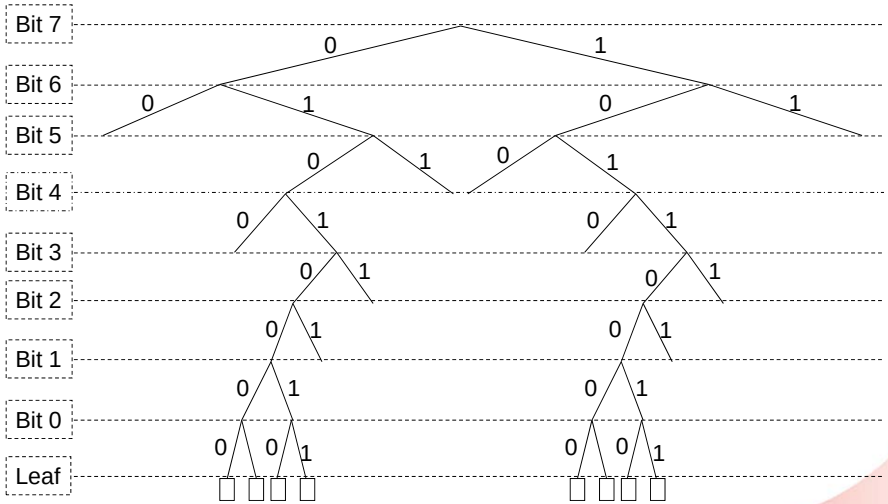


How to construct the tree representation of ASCII codes?

- The creation involves the following steps:
  a) to create eight levels from bit7, bit6, until leaf.
  b) to start with the single node at bit7
  c) to go down from one node by following two branches to a lower level. And, assign “0” to left-branch, and “1” to right branch.
  d) Repeat c) until reaching leaf level.



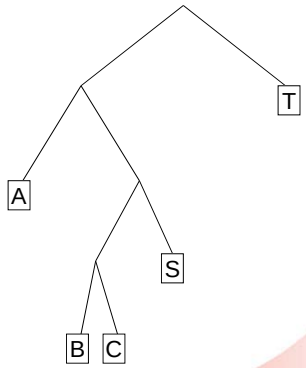
- Example:



What is a Huffman code?

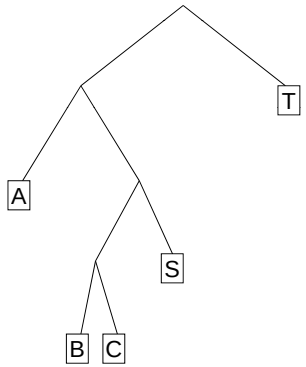
- Your answer:

  a)



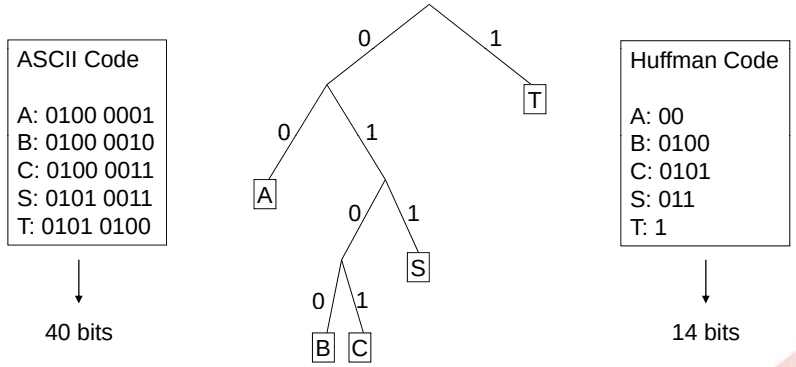
What is a Huffman code?

- A Huffman code refer to a code of variable bit-length.



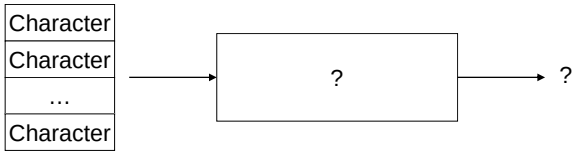
What is a Huffman code?

- Example:



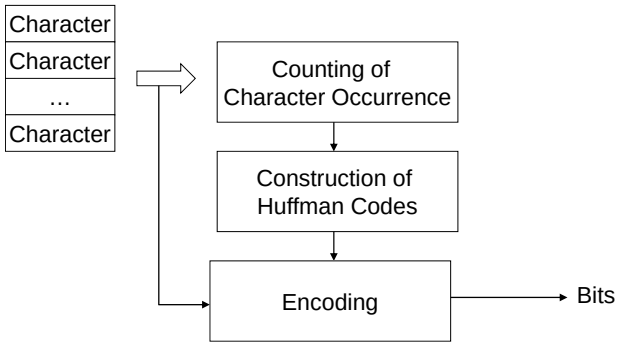
What is the procedure of data compression?

- Your answer:



What is the procedure of data compression?

- Data compression with Huffman codes involves the following steps:



How to construct Huffman codes?

- Algorithm:

  a) Start with the leaf nodes. Each node carries a weight, which is equal to the occurrence number of the associated character.

  b) Align the leaf nodes in the descending order from left to right.

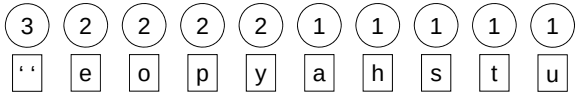
  c) Repeat the operation below until only one node remains at the top:
  - Find two adjacent nodes, of which the sum of weights is the smallest.
  - Group these two nodes to be under a new node.
  - Assign the sum of weights to this new node.
  - Activate this new node, and disable the two nodes under the new node.

How to construct Huffman codes?

- Example:

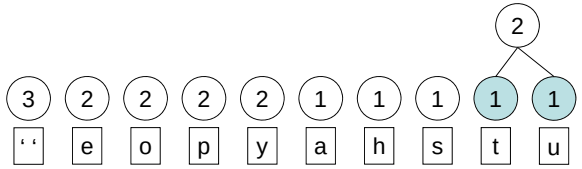
  Compress the text “happy to see you”

Iteration 1:



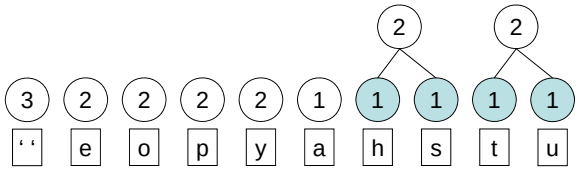
“happy to see you”

Iteration 2:



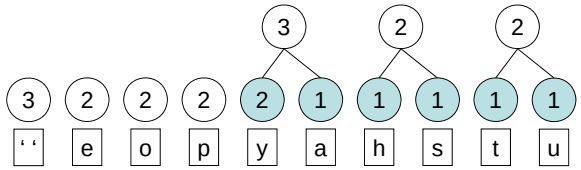
“happy to see you”

Iteration 3:



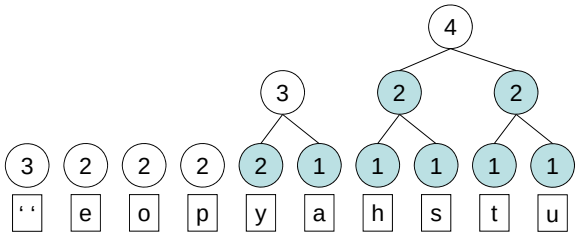
“happy to see you”

Iteration 4:



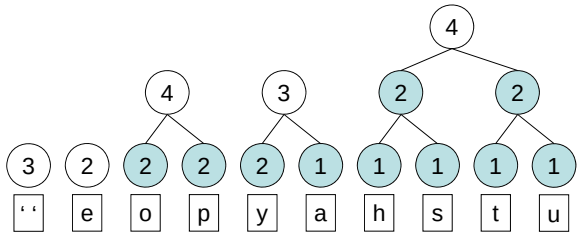
“happy to see you”

Iteration 5:



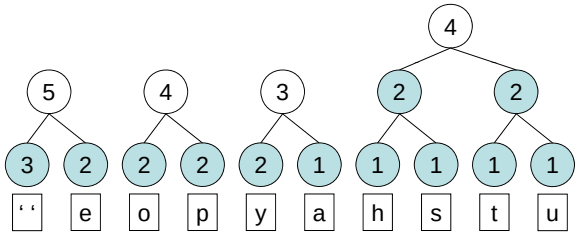
“happy to see you”

Iteration 6:



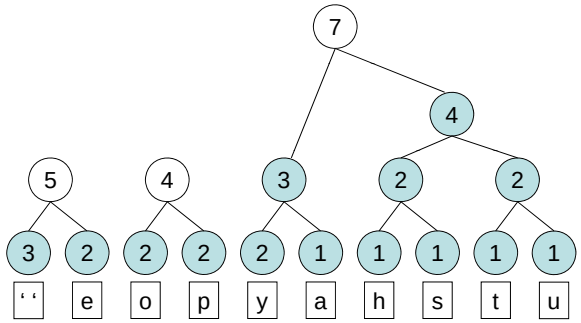
“happy to see you”

Iteration 7:



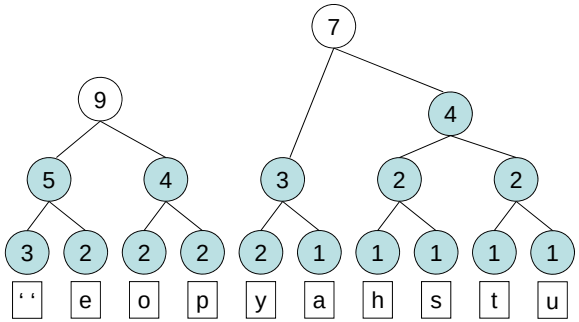
“happy to see you”

Iteration 8:



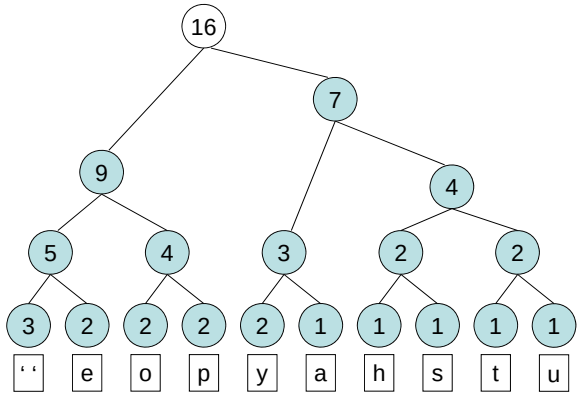
“happy to see you”

Iteration 9:



“happy to see you”

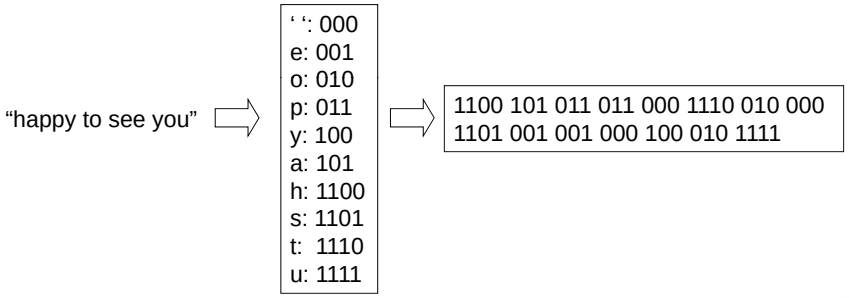
Iteration 10:



“happy to see you”

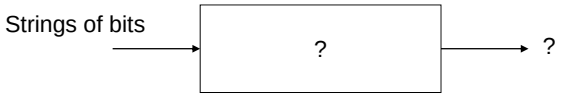
How to do encoding?

- Example: Compress the text “happy to see you”



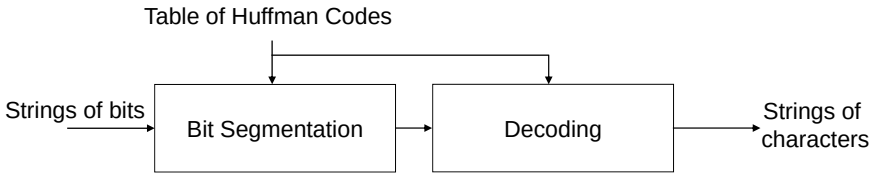
What is the procedure of data decompression?

- Your answer:



What is the procedure of data decompression?

- Data decompression involves the following steps:



How to do bit segmentation?

- Answer:

  The decompression involves the following steps:

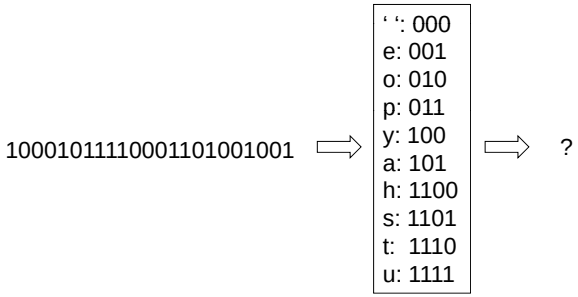
  a) Start from the first bit of the remaining string.

  b) Find the sub-string, which exists in the table of Huffman codes.

  c) Remove the sub-string and repeat.

How to do bit segmentation?

- Example: decompress “100010111100011010010 01”.



Q & A

Q & A

- What are the benefits of compressing data?
- In decompression with LZW algorithm, how to consider the special case, in which Symbol(thisC) does not exist at the current step?

```
…
while (there is still code to read)
{
        thisC = read in a code;
        if Symbol(thisC) does not exist,
                output “prevS+prevS[0]” ;
                add “prevS+prevS[0]” to dictionary;
        else
                …
}
```


Part 2:

Image Compression

Learning Objectives

- Image
- Compression
- GIF Algorithm
- JPEG Algorithm
- Q & A

Image

What is an digital image?

- Your answer:

  a)

What is a digital image?

- A digital image is a matrix of pixels, each of which contains a color.

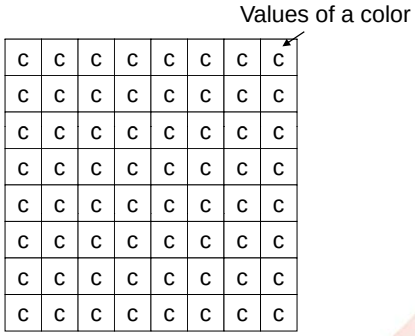
Values of a color

c	c	c	c	c	c	c	c
c	c	c	c	c	c	c	c
c	c	c	c	c	c	c	c
c	c	c	c	c	c	c	c
c	c	c	c	c	c	c	c
c	c	c	c	c	c	c	c
c	c	c	c	c	c	c	c
c	c	c	c	c	c	c	c

How to represent the values of a color?

- Your answer:

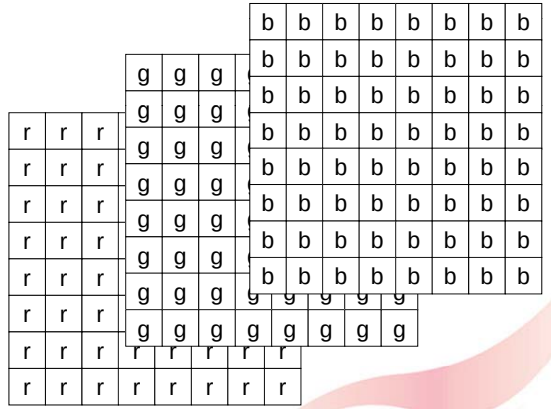
  a)



c	c	c	c	c	c	c	c
c	c	c	c	c	c	c	c
c	c	c	c	c	c	c	c
c	c	c	c	c	c	c	c
c	c	c	c	c	c	c	c
c	c	c	c	c	c	c	c
c	c	c	c	c	c	c	c
c	c	c	c	c	c	c	c

How to represent the values of a color?

- In general, a color is represented by three values, which are called tri-stimulus values, such as RGB, L*a*b, YCbCr, etc.



- Example 1:

  How to transform RGB into another three sets of values?

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.144 \\ -0.159 & -0.332 & 0.050 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} x \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

- Example 2:

  An image is represented by a matrix of pixels, each of which has a set of RGB values. If each value is encoded by one byte, what will be the size if the resolution of the image is 512x512?

Matrix of R Values: 512x512x8 (bits)
Matrix of G Values: 512x512x8 (bits)
Matrix of B Values: 512x512x8 (bits)

Total number of bits: 512x512x8x3 (bits)

- Example 3:

We have an image of the size of 8x8. The values at the pixels are shown below. What will be the size of this image?

b	b	b	b	b	b	b	b
b	b	r	r	r	r	b	b
b	b	r	r	r	r	b	b
b	b	r	r	r	r	b	b
b	b	g	g	g	g	b	b
b	b	g	g	g	g	b	b
b	b	g	g	g	g	b	b
b	b	b	b	b	b	b	b

Solution 1:

The size of the image is 8x8x8 (bits)

b	b	b	b	b	b	b	b
b	b	r	r	r	r	b	b
b	b	r	r	r	r	b	b
b	b	r	r	r	r	b	b
b	b	g	g	g	g	b	b
b	b	g	g	g	g	b	b
b	b	g	g	g	g	b	b
b	b	b	b	b	b	b	b

Solution 2:

b	b	b	b	b	b	b	b
b	b	r	r	r	r	b	b
b	b	r	r	r	r	b	b
b	b	r	r	r	r	b	b
b	b	g	g	g	g	b	b
b	b	g	g	g	g	b	b
b	b	g	g	g	g	b	b
b	b	b	b	b	b	b	b

Row 1: 8xb
Row 2: 2xb 4xr 2xb
Row 3: 2xb 4xr 2xb
Row 4: 2xb 4xr 2xb
Row 5: 2xb 4xg 2xb
Row 6: 2xb 4xg 2xb
Row 7: 2xb 4xg 2xb
Row 8: 8xb

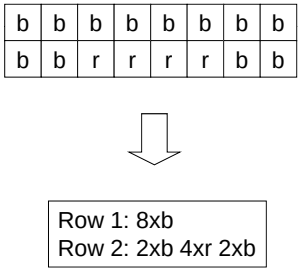
(2+6x6+2)x8 = 40x8 (bits)

Compression

What is image compression?

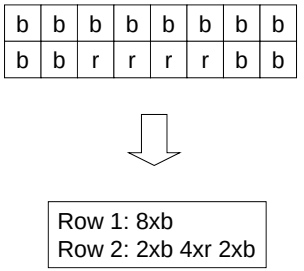
- Your answer:

  a)



What is image compression?

- By definition, image compression is to transform the original representation of an image into another form of representation with reduced size.

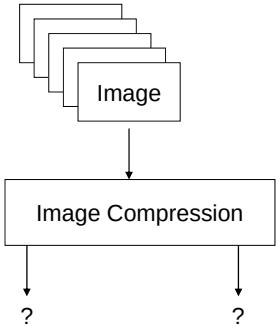


Why is there a need of doing image compression?

- Your answer:

  a)

  b)

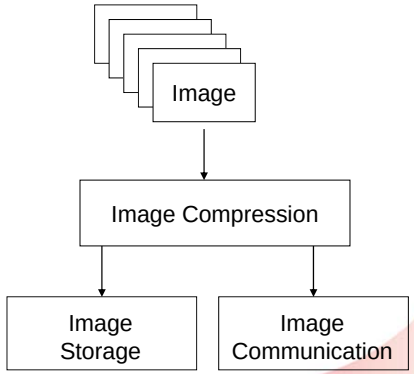


Why is there a need of doing image compression?

- In general, image compression is motivated by one of the following two reasons:

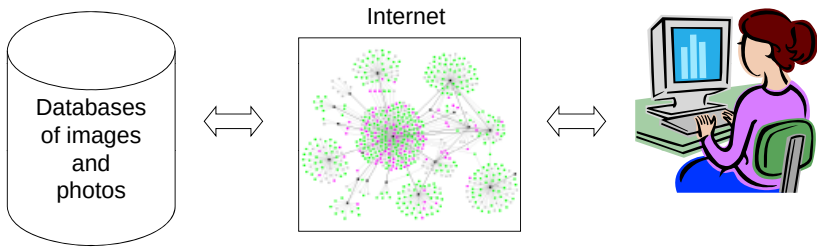
  a) image communication

  b) image storage



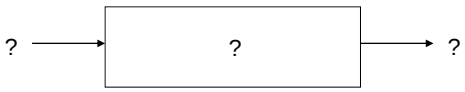
- Example 1: Image Storage.

- Example 2: Image Retrieval and Communication.



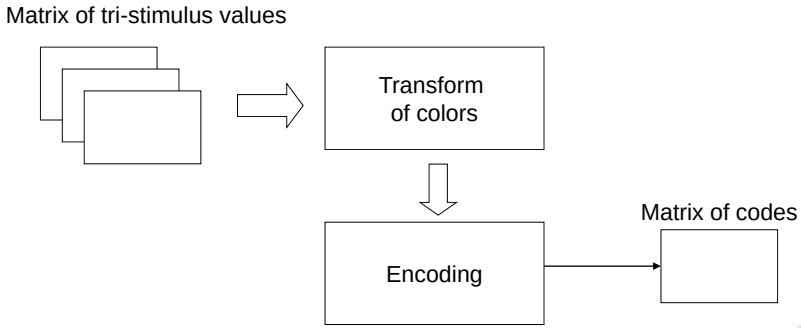
What is the general procedure of compressing an image?

- Your answer:



What is the general procedure of compressing an image?

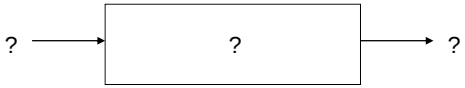
- In general, image compression involves two steps:



What is the general procedure of decompressing an image?

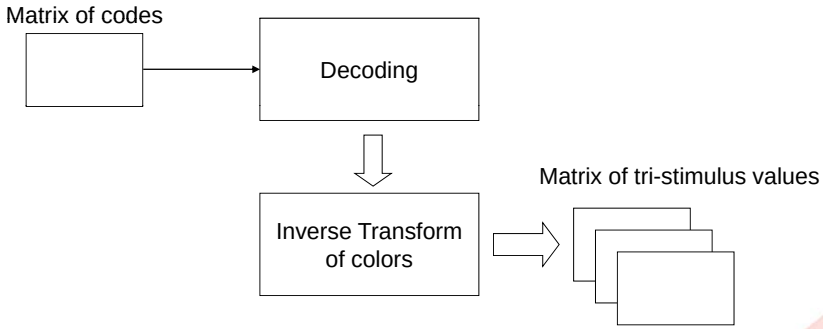
- Your answer:

  a)



What is the general procedure of decompressing an image?

- In general, image decompression also involves two steps:



GIF Algorithm

How good is GIF compression?

- Example:

Original: 116kb

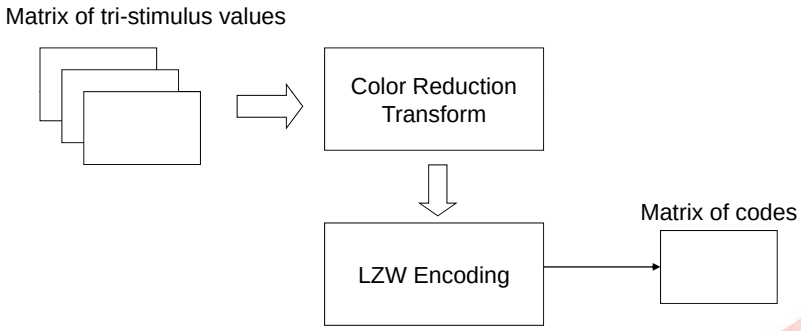
GIF with 256 colors: 24kb

GIF with 16 colors: 9.88kb

GIF with 8 colors: 6.44kb

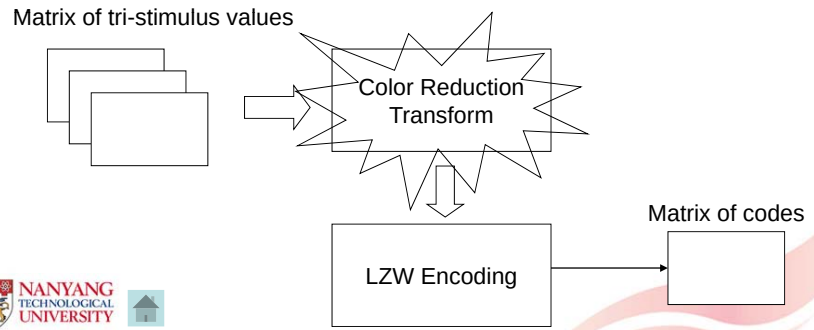
What is the procedure of image compression?

- Answer:



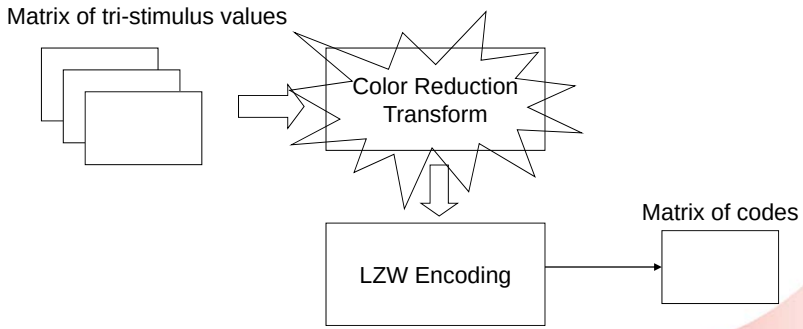
What's the purpose of reducing colors?

- Your answer:



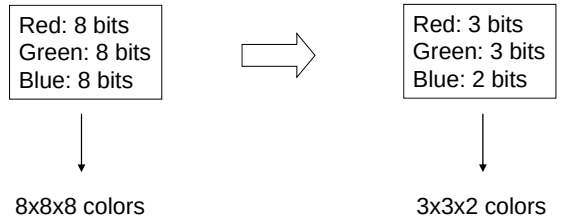
What's the purpose of reducing colors?

- The purpose of color reduction is to reduce the number of colors to 256 colors in maximum so that each color is represented by a byte.



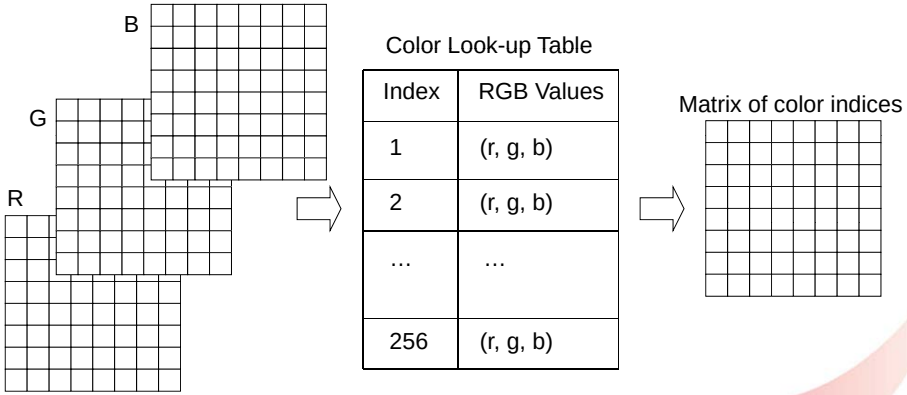
How to reduce the number of colors?

- Example 1:



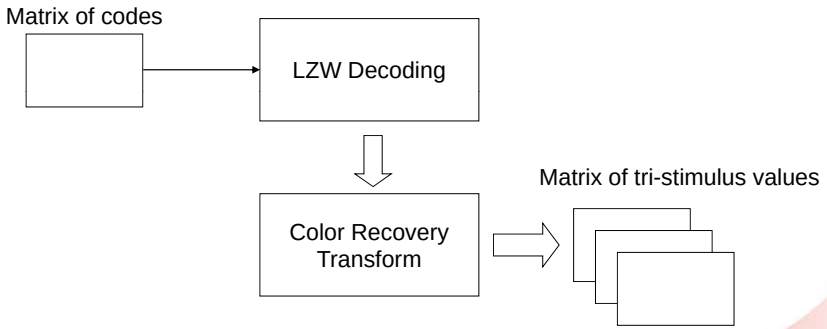
How to reduce the number of colors?

- Example 2:



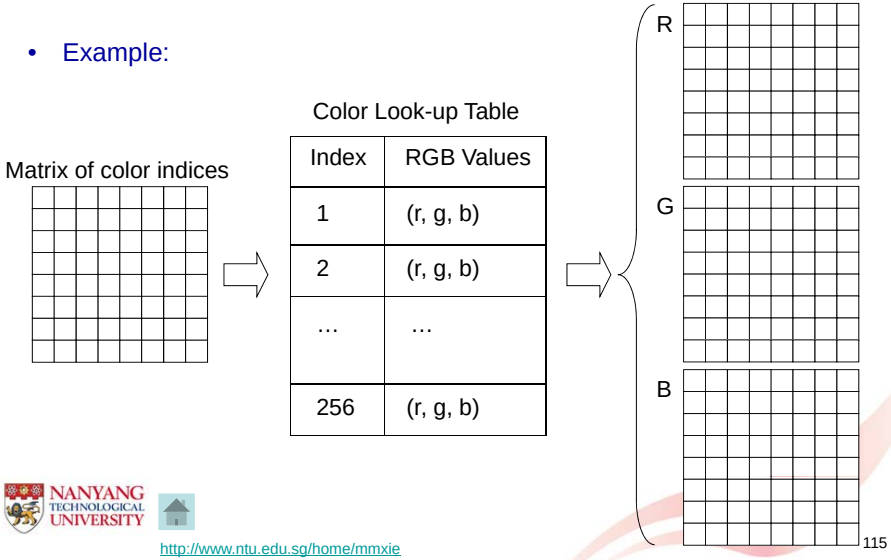
What is the procedure of image decompression

- Answer:



How to recover colors?

- Example:



JPEG Algorithm

How good is JPEG compression?

- Example:

Original: 116kb

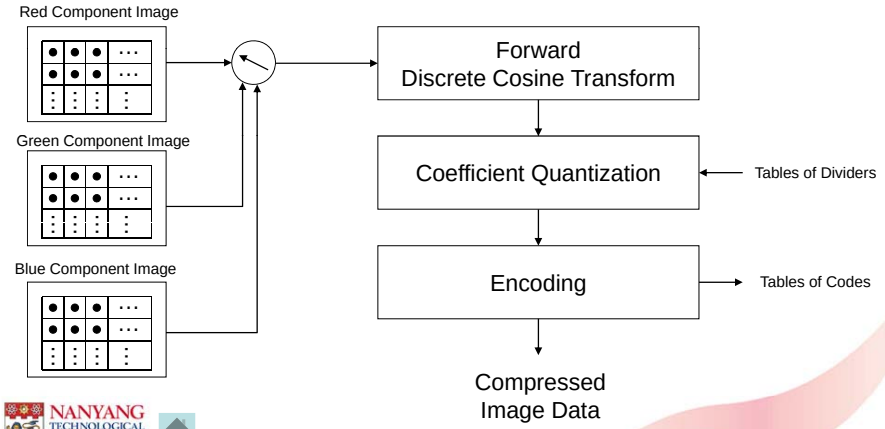
Quality 90%: 12.9kb

Quality 40%: 4.6kb

Quality 10%: 1.95kb

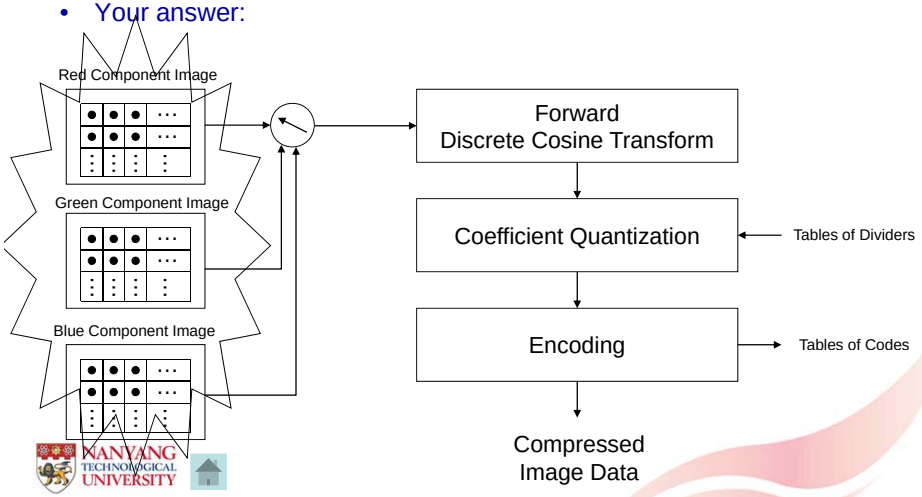
What is the procedure of image compression?

- Answer:



How is a component image organized?

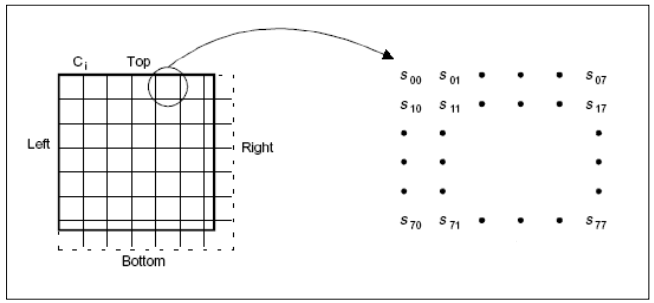
- Your answer:



How is a component image organized?

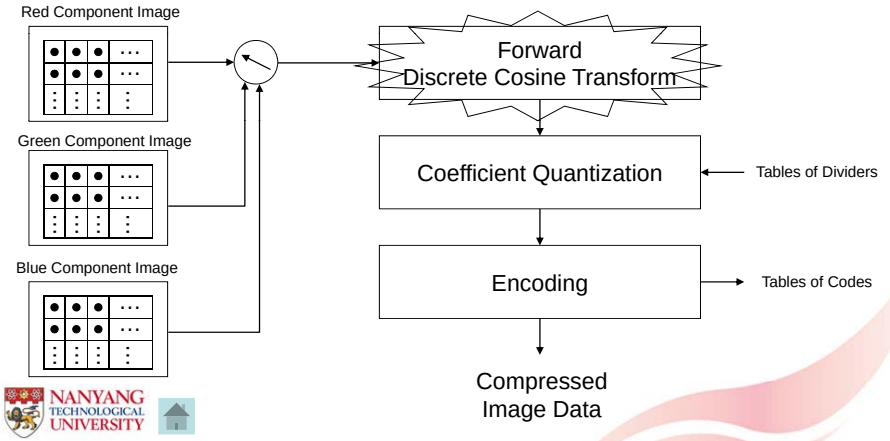
- Answer:

Each component image is divided into blocks of 8x8 pixels.



How to perform forward discrete cosine transform?

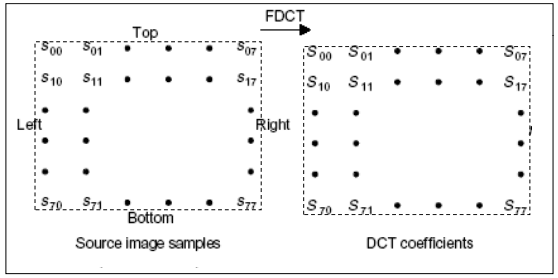
- Your answer:



How to perform forward discrete cosine transform?

- Answer:

It transforms a matrix of 8x8 values into another matrix of 8x8 coefficients.



How to perform forward discrete cosine transform?

- Answer (continued):

And, the equation of FDCT is:

$$S_{vu} = \frac{1}{4} C_u C_v \sum_{x=0}^{7} \sum_{y=0}^{7} s_{yx} \cos\left(\frac{(2x+1)u\pi}{16}\right) \cos\left(\frac{(2y+1)v\pi}{16}\right)$$

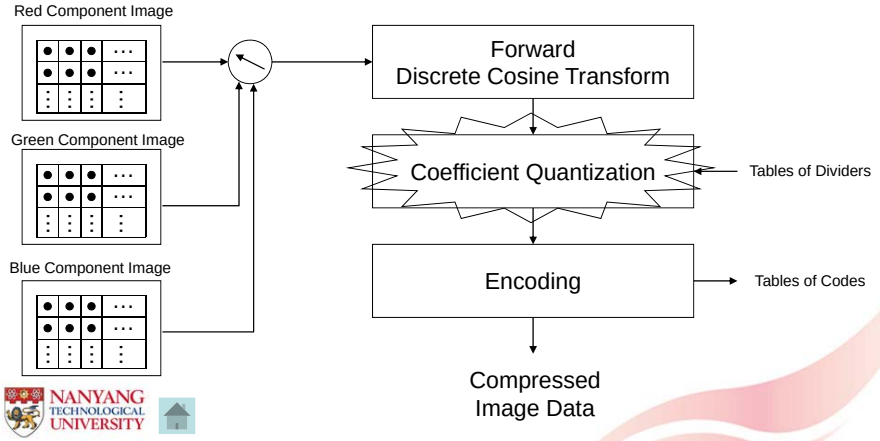
where

$$C_u = C_v = \frac{1}{\sqrt{2}} \quad \text{if u} = 0 \text{ and v} = 0$$

$$C_u = C_v = 1 \quad \text{otherwise}$$

How to perform coefficient quantization?

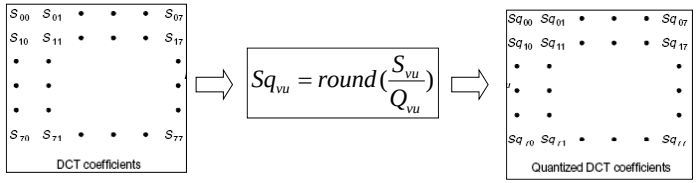
- Your answer:



How to perform coefficient quantization?

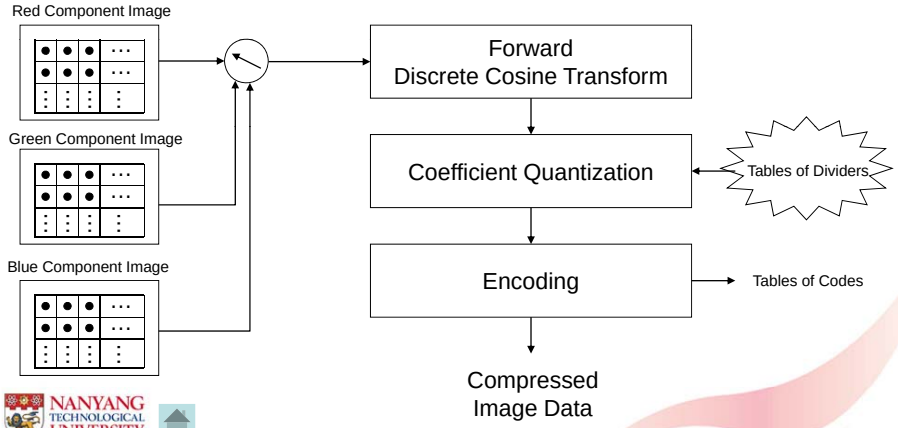
- Answer:

Each coefficient is divided by a value, which is called "quantizer step size".



How to determine the table of quantizer step sizes?

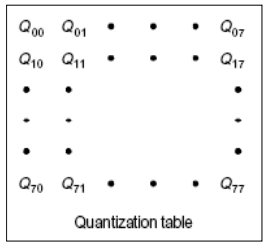
- Your answer:



How to determine the table of quantizer step sizes?

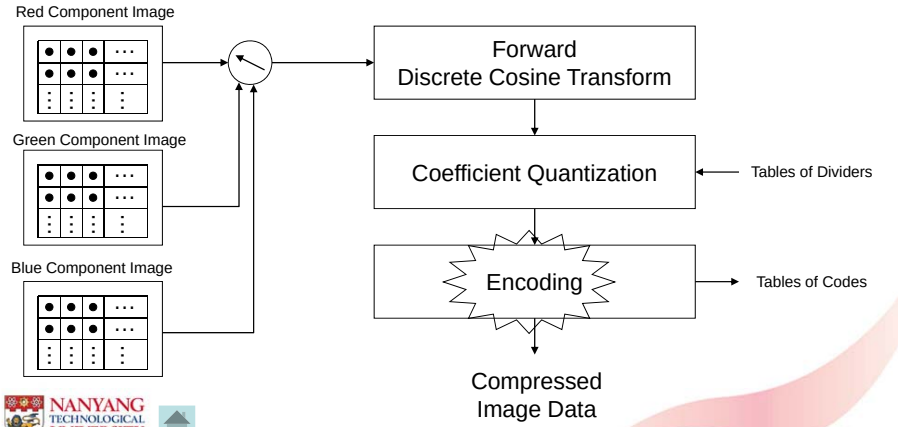
- Answer:

It depends on the desired quality of uncompressed image.



How to performance encoding?

- Your answer:



How to performance encoding?

- Answer:

  a) First of all, the matrix of 8x8 quantized coefficients is transformed into a string of characters.

  b) The string of characters is compressed by Huffman algorithm.

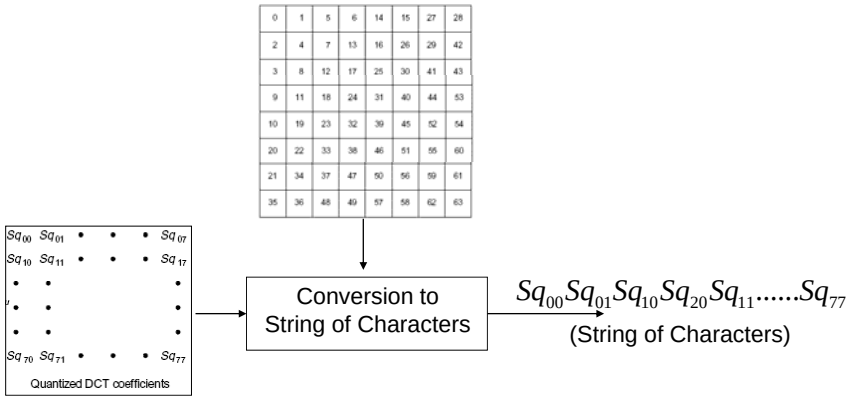
How to performance encoding?

- Answer (continued):

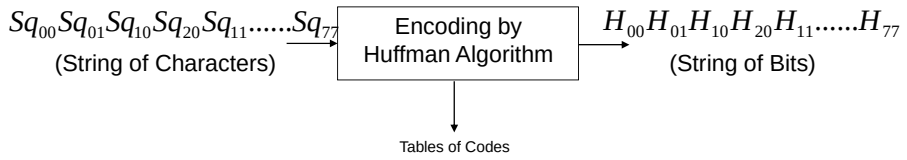
  The sequential orders of the 8x8 coefficients are determined by the following matrix:

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

Answer (continued) :

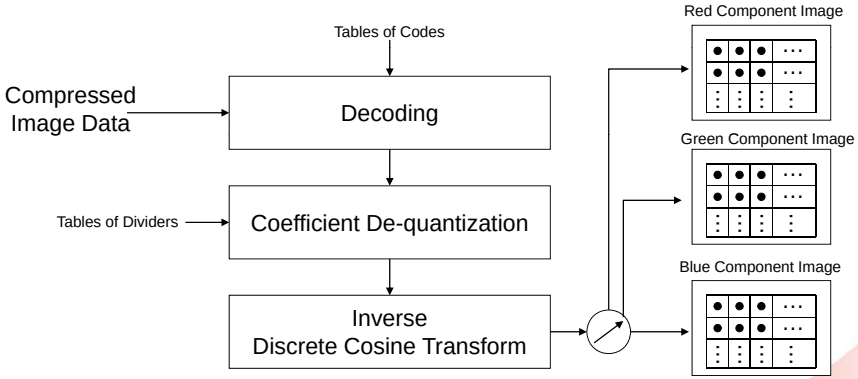


Answer (continued) :

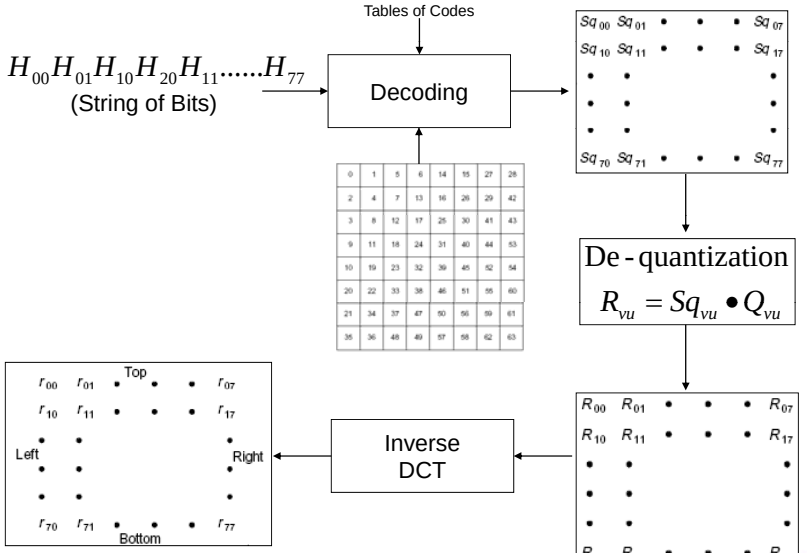


What is the procedure of image decompression?

- Answer:

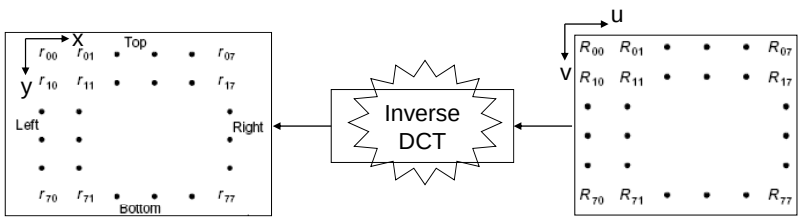


Answer (continued):



How to perform inverse discrete cosine transform?

- Your answer:



How to perform inverse discrete cosine transform?

- Answer:

The equation of inverse discrete cosine transform is:

$$r_{yx} = \frac{1}{4}\sum_{u=0}^{7}\sum_{v=0}^{7} C_u C_v R_{vu} \cos\left(\frac{(2x+1)u\pi}{16}\right)\cos\left(\frac{(2y+1)v\pi}{16}\right)$$

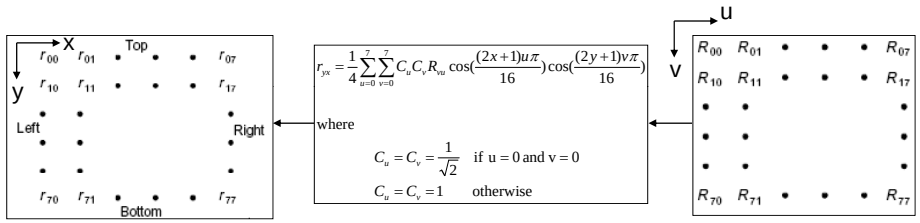
where

$$C_u = C_v = \frac{1}{\sqrt{2}} \quad \text{if u = 0 and v = 0}$$

$$C_u = C_v = 1 \quad \text{otherwise}$$

How to perform inverse discrete cosine transform?

- Answer (continued):



Q & A

Q & A

- In GIF algorithm, if a variable varies between 0 to 255, how to scale its range down to the interval [0 7]?
- Will GIF algorithm cause the loss of colors?
- Will JPEG algorithm cause the loss of colors?

Part 3:

Video Compression

Learning Objectives

- Video
- Compression
- MPEG Algorithm
- Q & A

What is a digital video?

- Your answer:

  a)

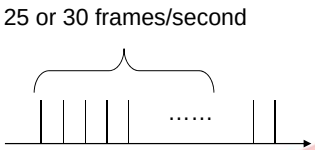
  b)

What is a digital video?

- Depending on standard, a digital video is:

  a) a sequence of digital images at 25 frames/second in PAL format

  b) a sequence of digital images at 30 frames/second in NTSC format



What is a frame in a digital video?

- Your answer:

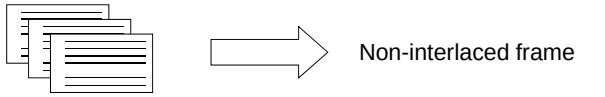
  a)

  b)

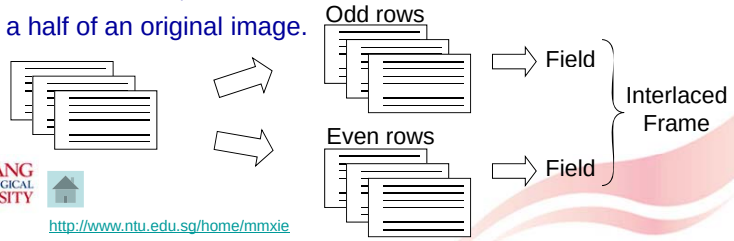
What is a frame in a digital video?

- In digital video, there are two types of frames:

  a) non-interlaced frames, each of which just corresponds a matrix of tri-stimulus values.



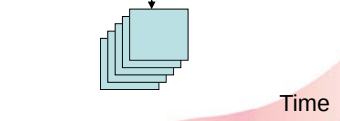
  b) interlaced frames, each of which consists of two fields. Each field is a half of an original image.



How is a video represented?

- Your answer:

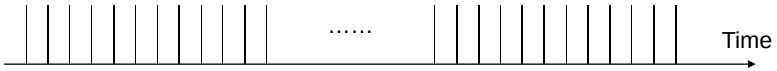
  a)



How is a video represented?

- A video can be represented by a set of images with time indices.

{Image(k), k = 1, 2, 3, …, N}



Compression

Why to compress a digital video?

- Example:

  In PAL format, the size of an image is 576 (rows) x 720 (columns).

  If each pixel contains a RGB color and each color component occupies one byte, what will be the size of a video in a second?

576x720x25x(8 + 8 + 8) = 248.832 Mbits/second

Why to compress a digital video?

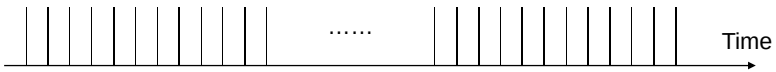
- Video compression is necessary not only for transmission, but also for storage/retrieval.

Is a video compressible?

- Your answer:

  a)

  b)

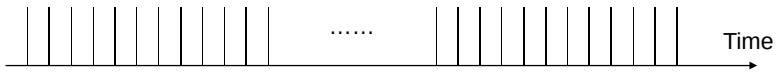


Is a video compressible?

- In general, a video is compressible because:

  a) there exists intra-frame redundancy, and also

  b) there is inter-frame redundancy



Is a video compressible?

- Example: Intra-frame redundancy.

b	b	b	b	b	b	b	b
b	b	r	r	r	r	b	b
b	b	r	r	r	r	b	b
b	b	r	r	r	r	b	b
b	b	g	g	g	g	b	b
b	b	g	g	g	g	b	b
b	b	g	g	g	g	b	b
b	b	b	b	b	b	b	b

Is a video compressible?

- Example 2: Inter-frame Redundancy.

Image(k)

b	b	b	b	b	b	b	b
b	b	r	r	r	r	b	b
b	b	r	r	r	r	b	b
b	b	r	r	r	r	b	b
b	b	g	g	g	g	b	b
b	b	g	g	g	g	b	b
b	b	g	g	g	g	b	b
b	b	b	b	b	b	b	b

Image(k+1)

b	b	b	b	b	b	b	b
b	b	b	b	b	b	b	b
b	b	b	r	r	r	r	b
b	b	b	r	r	r	r	b
b	b	b	r	r	r	r	b
b	b	b	g	g	g	g	b
b	b	b	g	g	g	g	b
b	b	b	g	g	g	g	b

MPEG Algorithms

How many versions of MPEG algorithm are there?

- Your answer:

  a)

  b)

  c)

  d)

How many MPEG standards are there?

- There are:

  a) MPEG-1 for VCD at 1.5 Mbits/second

  b) MPEG-2 for VCD at 3 – 15 Mbits/second

  c) MPEG-3 for DVD

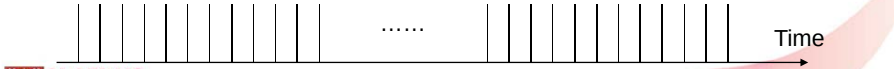
  d) MPEG-4, MPEG-7 , … (e.g. Video Phone)

How to exploit intra-frame redundancy to compress a video?

- Your answer:

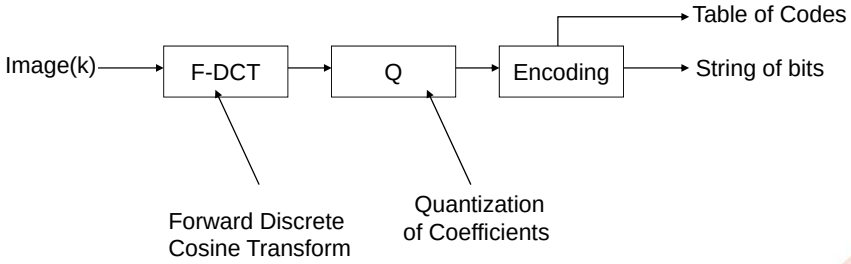
  a)

b	b	b	b	b	b	b	b
b	b	r	r	r	r	b	b
b	b	r	r	r	r	b	b
b	b	r	r	r	r	b	b
b	b	g	g	g	g	b	b
b	b	g	g	g	g	b	b
b	b	g	g	g	g	b	b
b	b	b	b	b	b	b	b



How to exploit intra-frame redundancy to compress a video?

- A common solution is to employ JPEG algorithm.



How to exploit inter-frame redundancy to compress a video?

- Your answer:

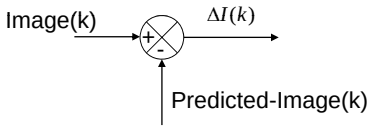
  a)

b	b	b	b	b	b	b	b
b	b	r	r	r	r	b	b
b	b	r	r	r	r	b	b
b	b	r	r	r	r	b	b
b	b	g	g	g	g	b	b
b	b	g	g	g	g	b	b
b	b	g	g	g	g	b	b
b	b	b	b	b	b	b	b

b	b	b	b	b	b	b	b
b	b	b	b	b	b	b	b
b	b	b	r	r	r	r	b
b	b	b	r	r	r	r	b
b	b	b	r	r	r	r	b
b	b	b	g	g	g	g	b
b	b	b	g	g	g	g	b
b	b	b	g	g	g	g	b

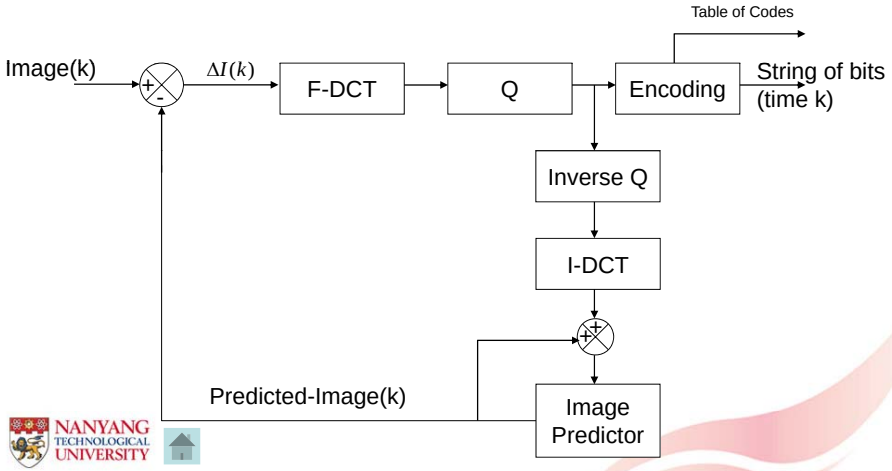
How to exploit inter-frame redundancy to compress a video?

- If we can predict Image(k+1) from Image(k), then the difference between Image(k+1) and Predicted-Image(k+1) will contain much less information.



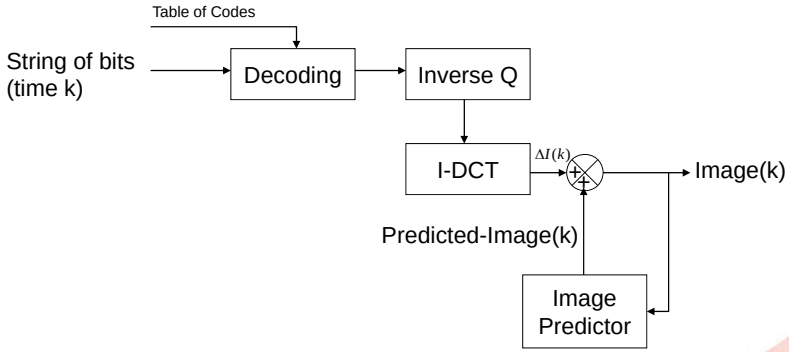
What is the procedure of video compression?

- Answer:



What is the procedure of video decompression?

- Answer:



Q & A

Q & A

- What is intra-frame redundancy?
- What is inter-frame redundancy?
- In MPEG algorithm, how to predict an image?

End