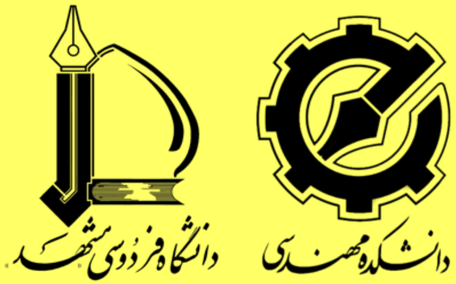
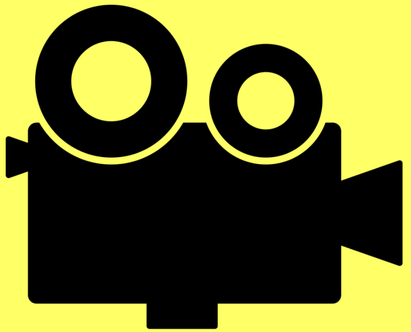


MULTIMEDIA SYSTEMS



AFSHIN ALAGHEHBAND
DEPARTEMANT OF COMPUTER ENGINEERING
FERDOWSI UNIVERSITY OF MASHHAD
>> SEMESTER-2, 1400-1401

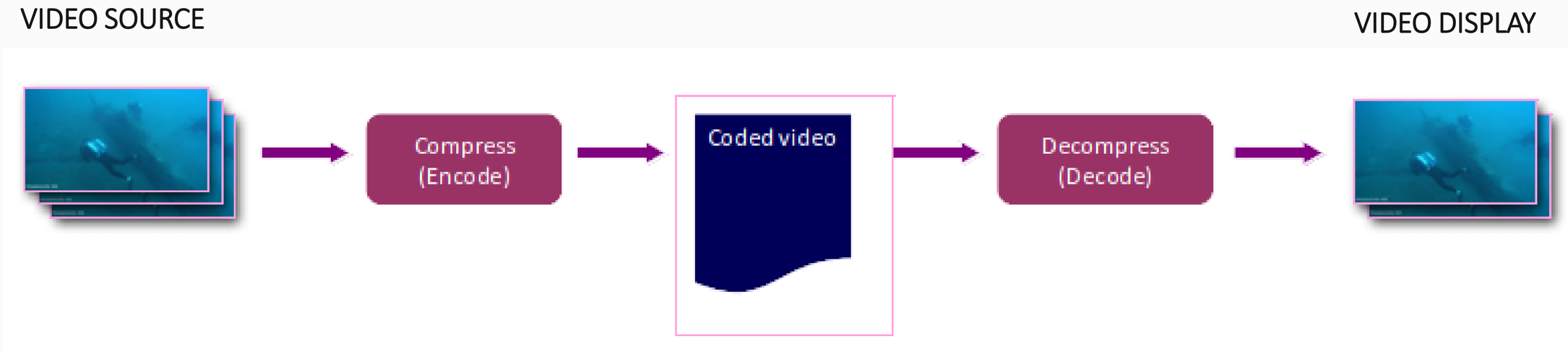
VIDEO CODEC

Project

Outline

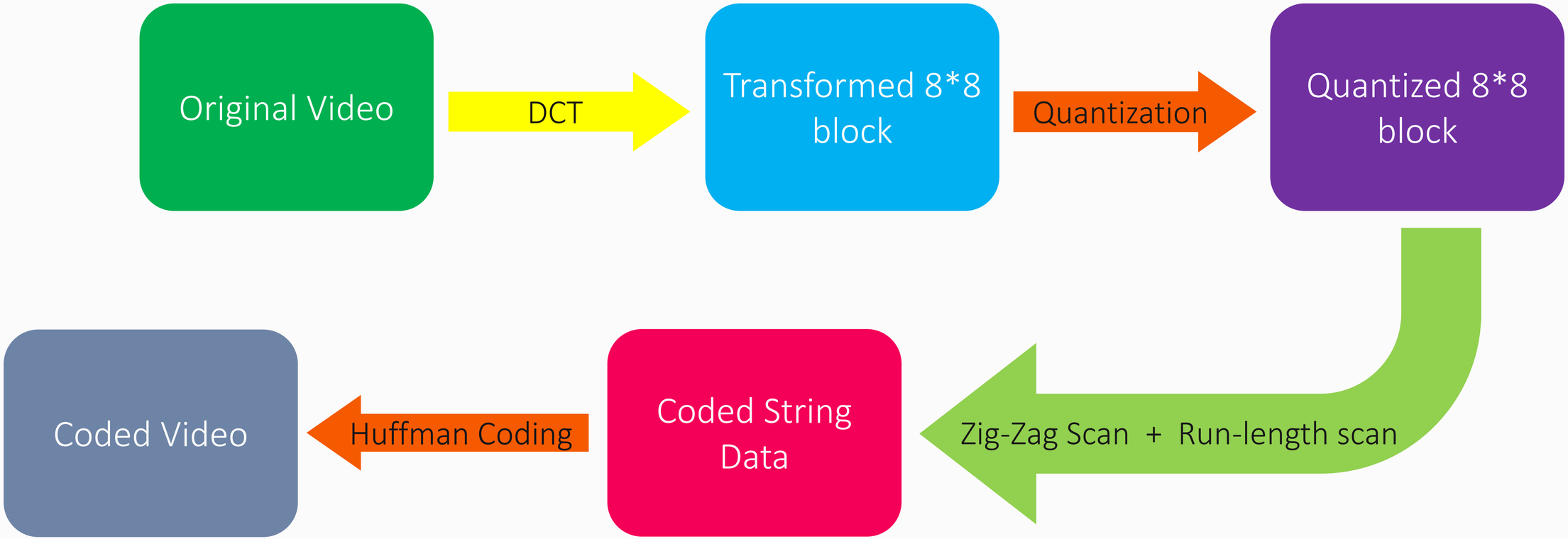
- Video CODEC
- Encoder
- Discrete Cosine Transform
- Quantization
- Zig-Zag Scan
- Run Length Scan
- Huffman Coding
- Decoder
- Comprehensive View
- Characteristics of Video
- Frame Difference
- Motion Estimation
- Frame Types
- Project

Video CODEC

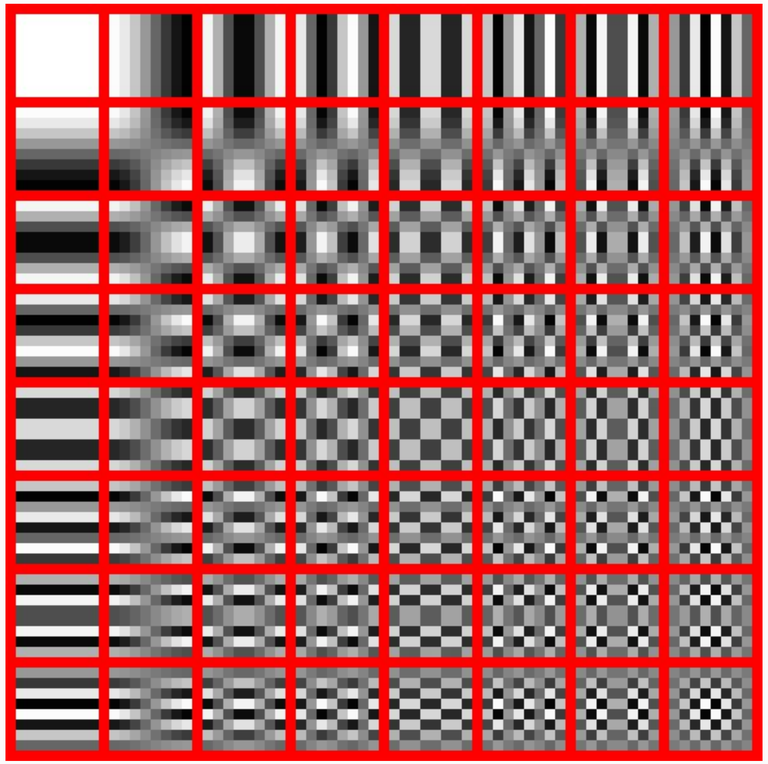


Encoder + Decoder = Codec

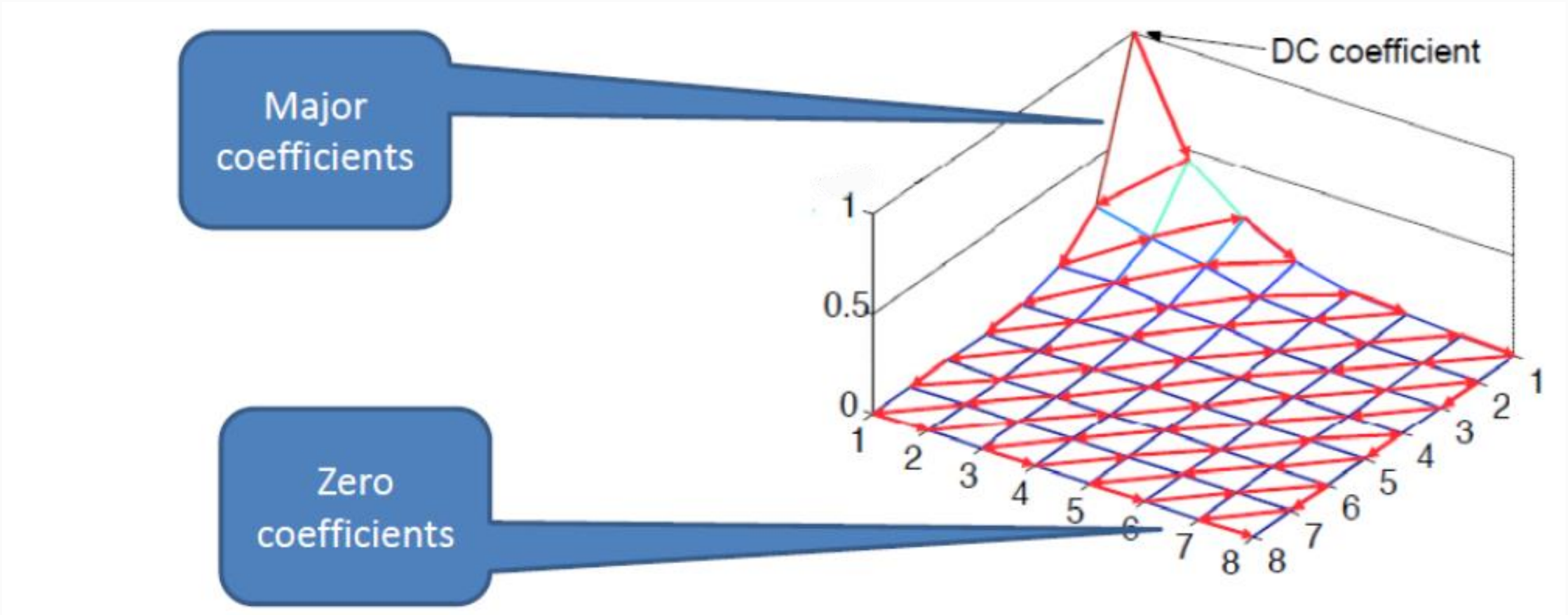
Encoder


Original Video
DCT
Transformed 8*8 block
Quantization
Quantized 8*8 block
Zig-Zag Scan + Run-length scan
Coded String Data
Huffman Coding
Coded Video

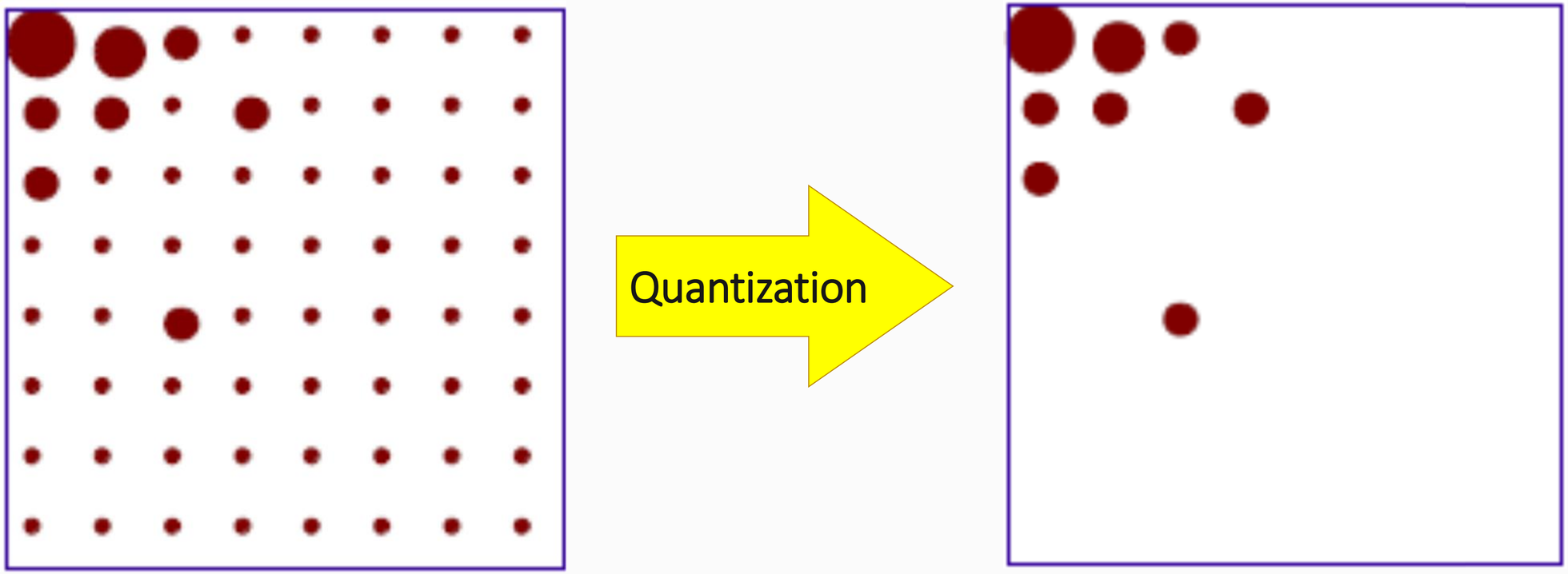
Discrete Cosine Transform



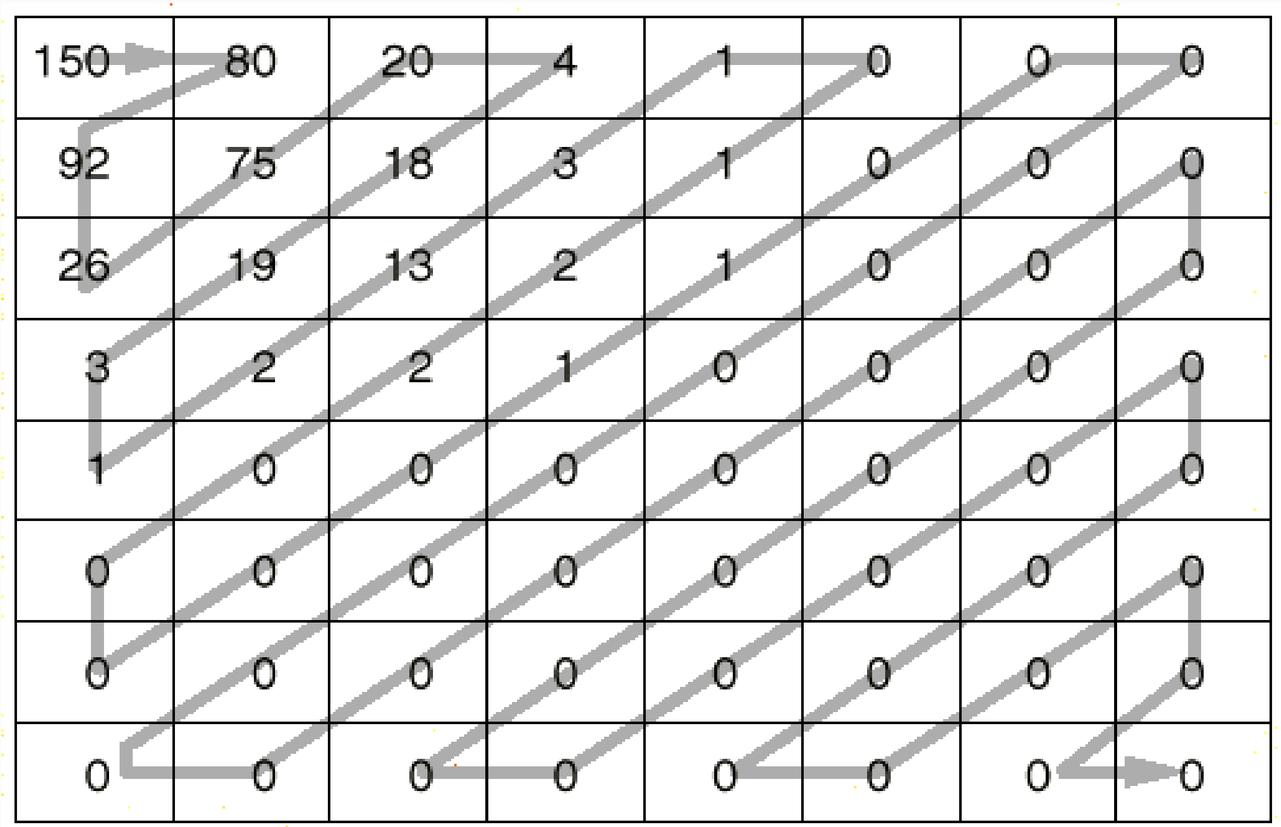
Discrete Cosine Transform



Quantization



Zig-Zag Scan


150 80 20 4 1 0 0 0
92 75 18 3 1 0 0 0
26 19 13 2 1 0 0 0
3 2 2 1 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

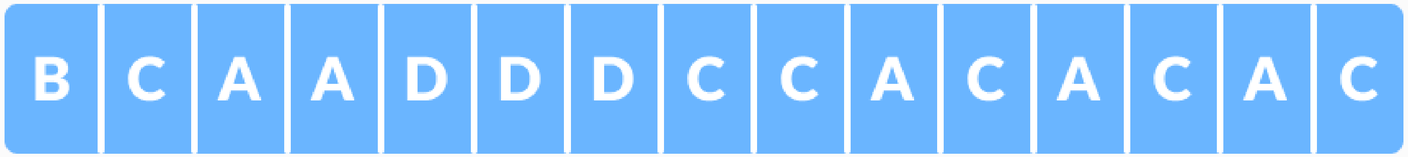
Run Length Scan

61	13	7	0	0	0	0	0
12	11	6	1	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	0	0	0	0	0	0	0

{61,0-13,0-12,1-11,0-7,1-6,0-1,4-1,EOB}

Huffman Coding

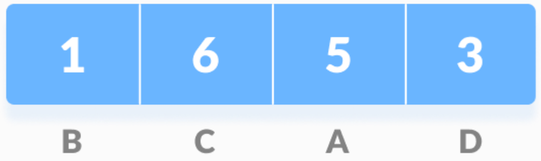
- lossless compression
- Audio & Video



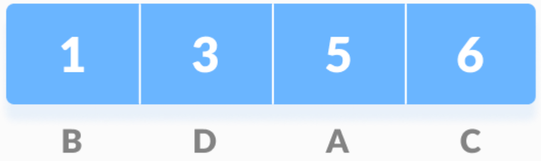
- Total bits = 8 * 15 = 120
- Total bits after encoding = 75

Huffman Coding

- Frequency of string



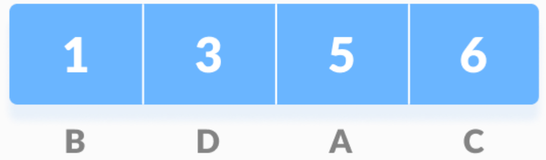
- Characters sorted according to the frequency
  - Sort the characters in increasing order of the frequency. These are stored in a priority queue Q

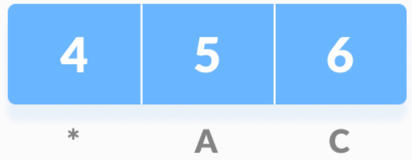


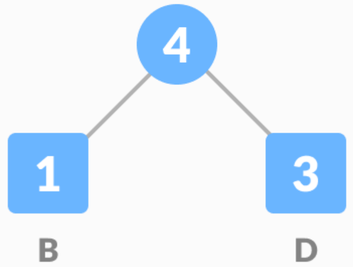
- Make the Huffman coding tree
  - Make each unique character as a leaf node

Huffman Coding

- Create an empty node Z. Assign the minimum frequency to the left child of Z and assign the second minimum frequency to the right child of Z. Set the value of the Z as the sum of the above two minimum frequencies.
- Remove these two minimum frequencies from Q and add the sum into the list of frequencies (* denote the internal nodes in the figure above).
- Insert node Z into the tree.

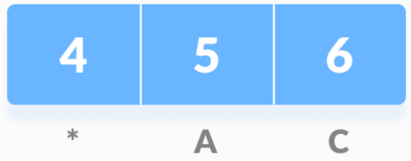


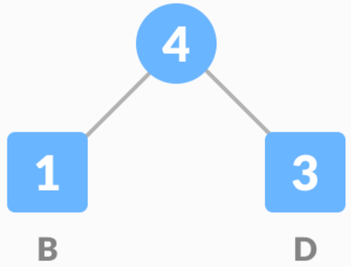


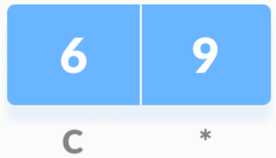


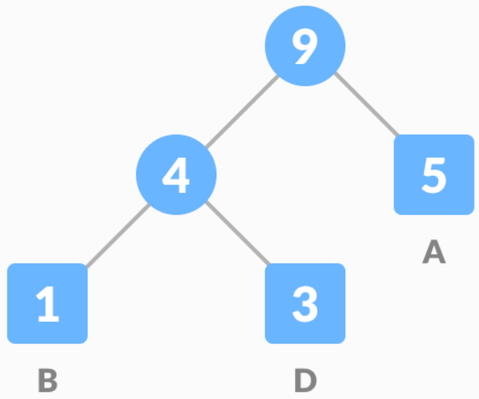
Huffman Coding

- Repeat 3 for all the characters

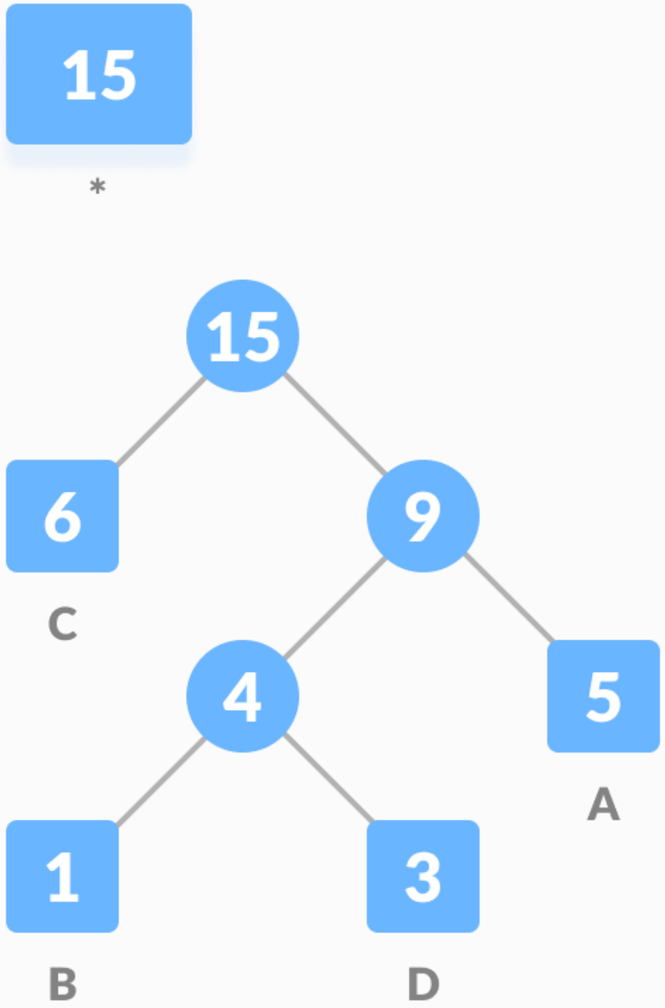








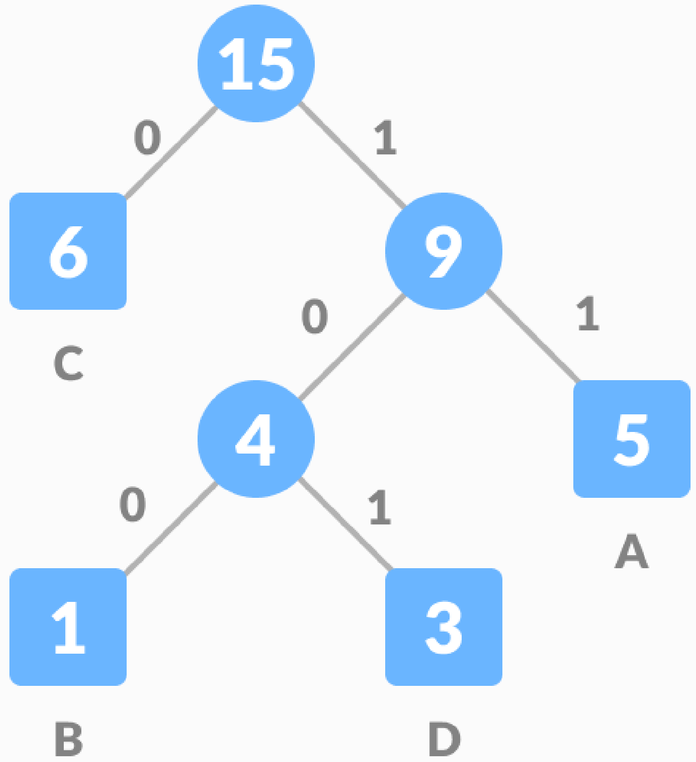
Huffman Coding



Huffman Coding

- For each non-leaf node, assign 0 to the left edge and 1 to the right edge.

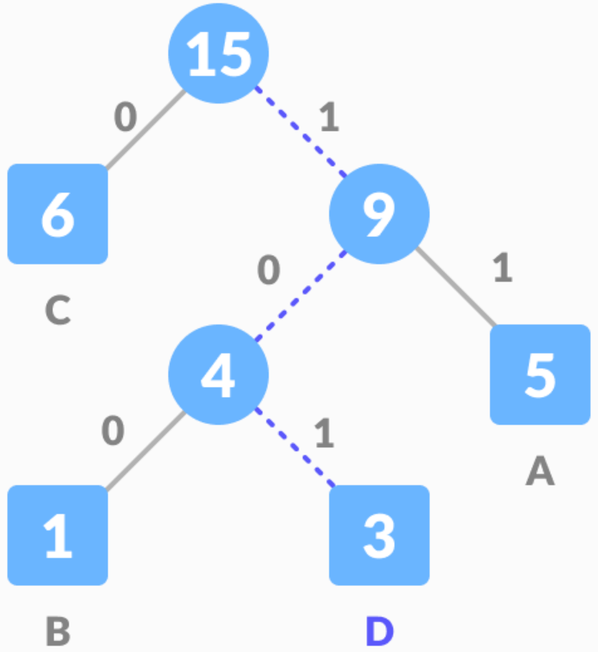
Character	Frequency	Code	Size
A	5	11	5*2 = 10
B	1	100	1*3 = 3
C	6	0	6*1 = 6
D	3	101	3*3 = 9
4 * 8 = 32 bits	15 bits		28 bits



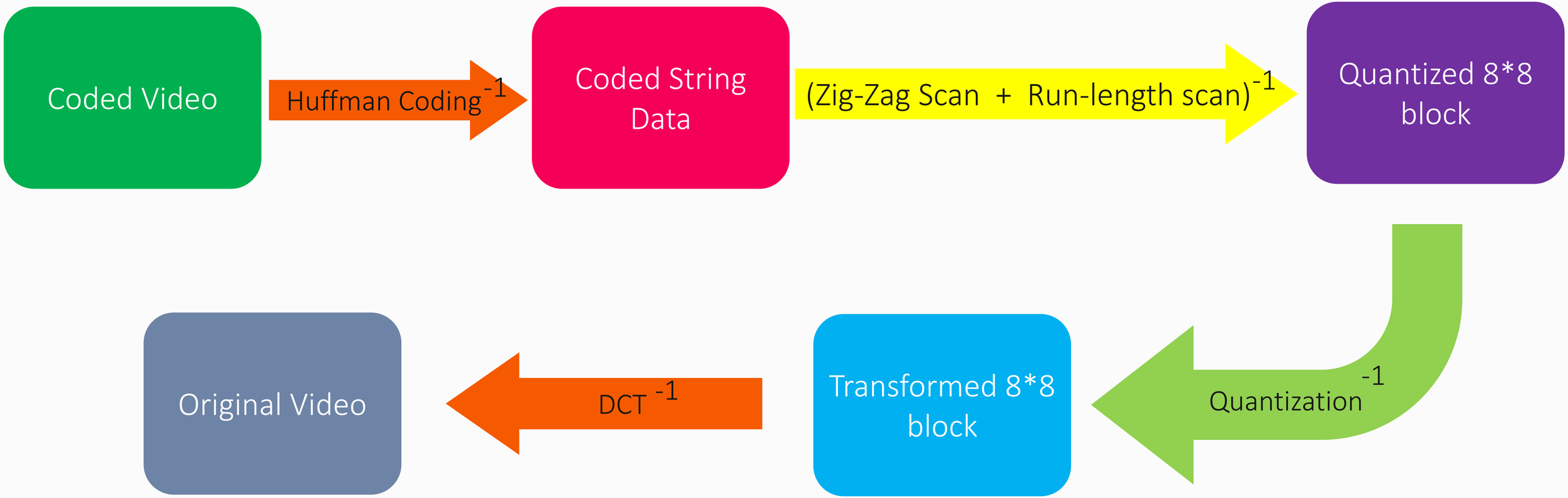
Huffman Decoding

- 1000111110110100110110110

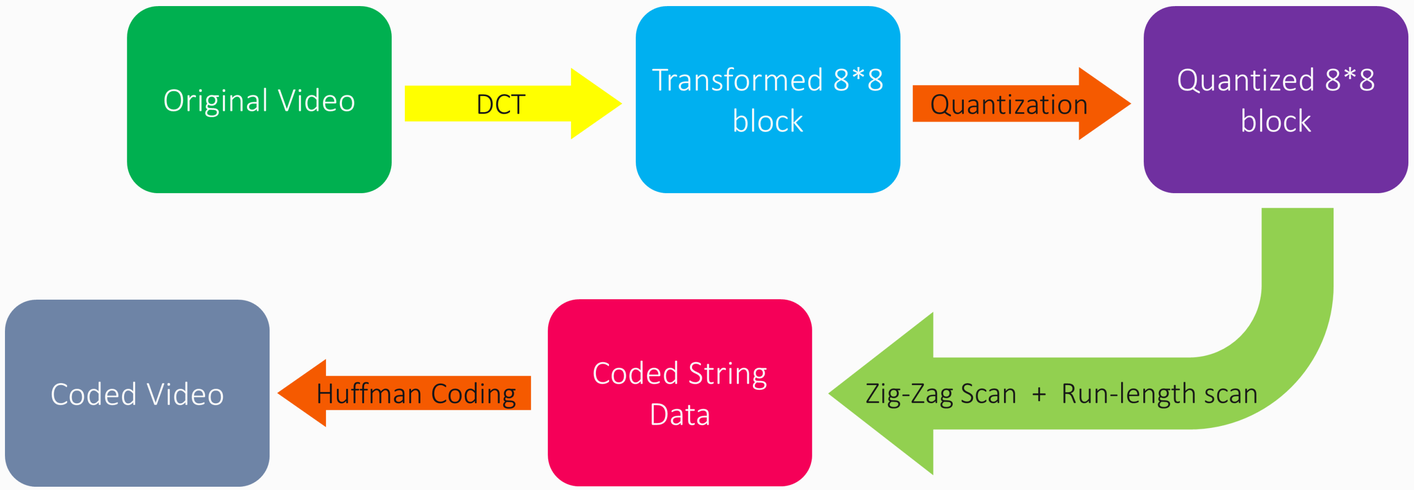
B	C	A	A	D	D	D	C	C	A	C	A	C	A	C



Decoder

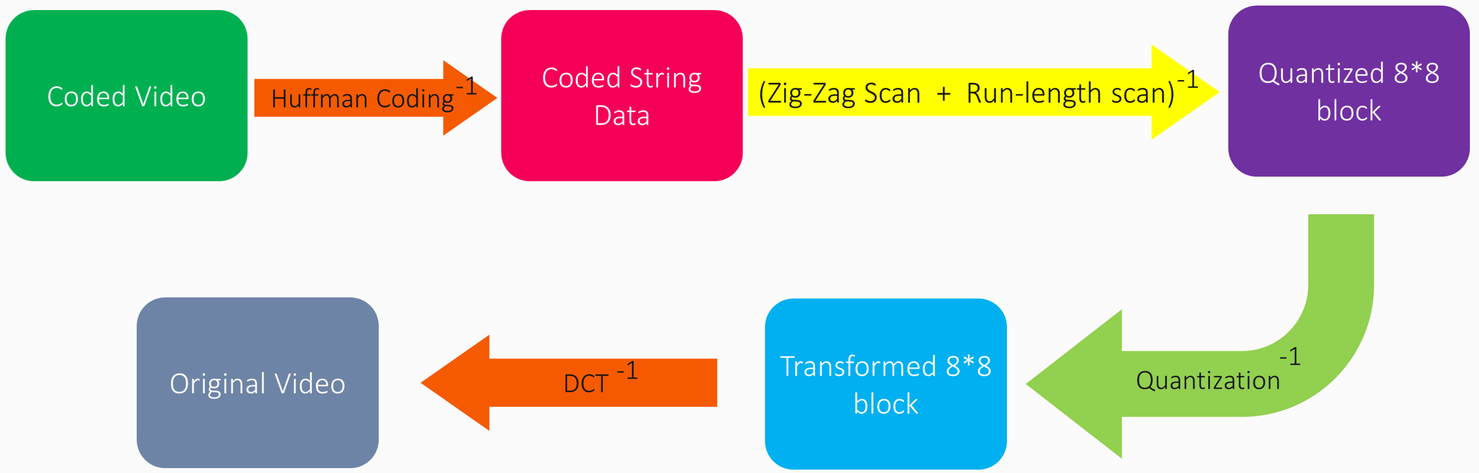

Coded Video
Huffman Coding -1
Coded String Data
(Zig-Zag Scan + Run-length scan) -1
Quantized 8*8 block
Quantization -1
Transformed 8*8 block
DCT -1
Original Video

Comprehensive View

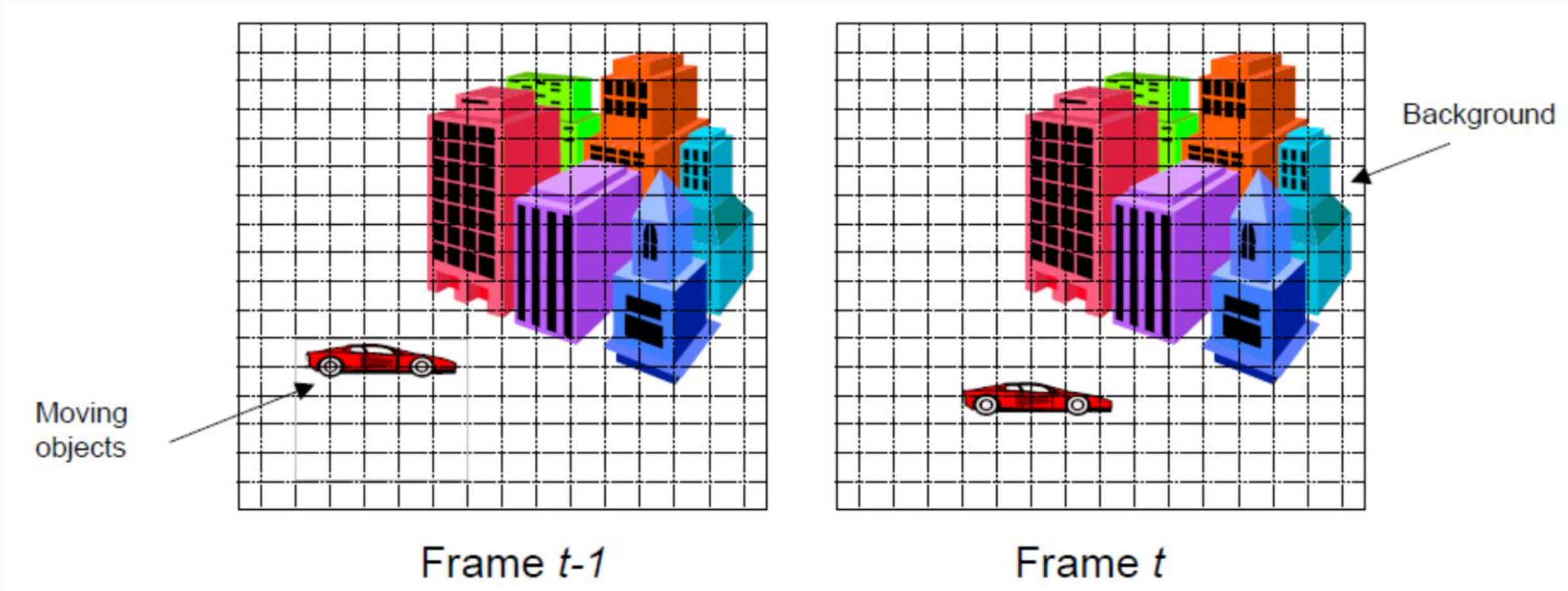


Encoder

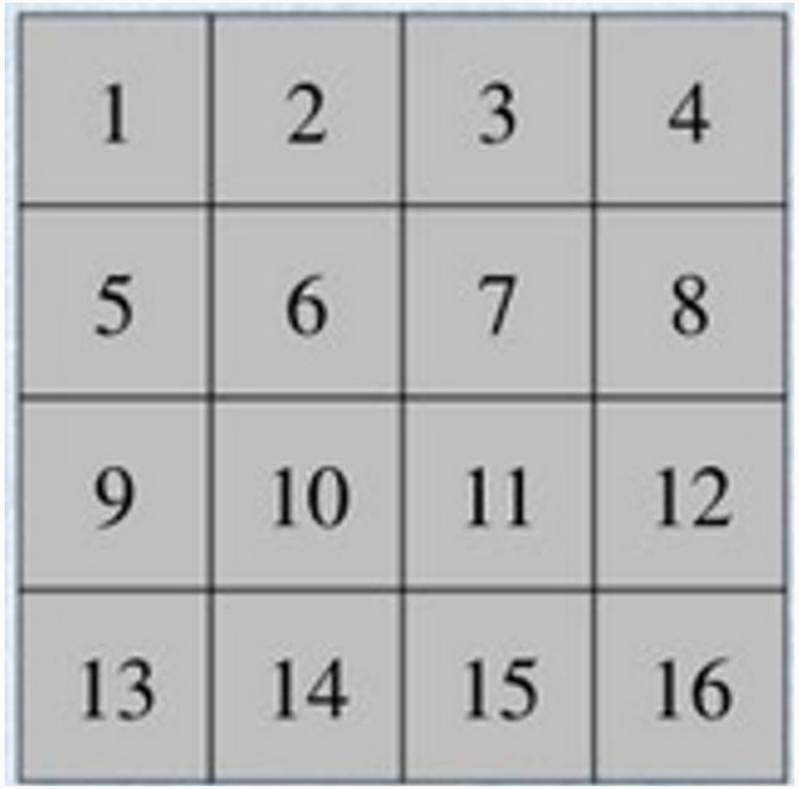
Decoder



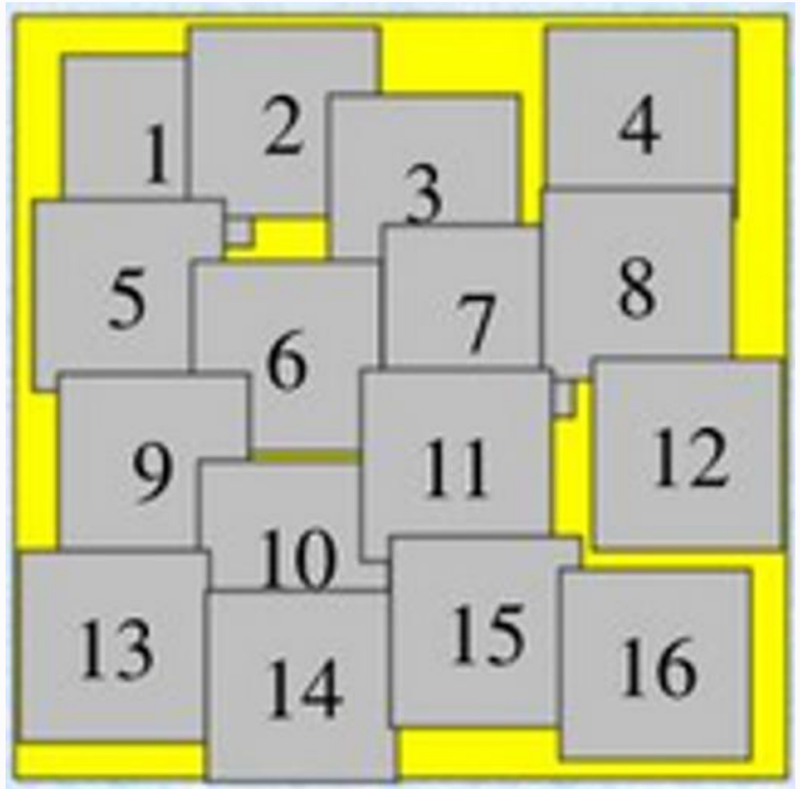
Characteristics of Video



Frame Difference

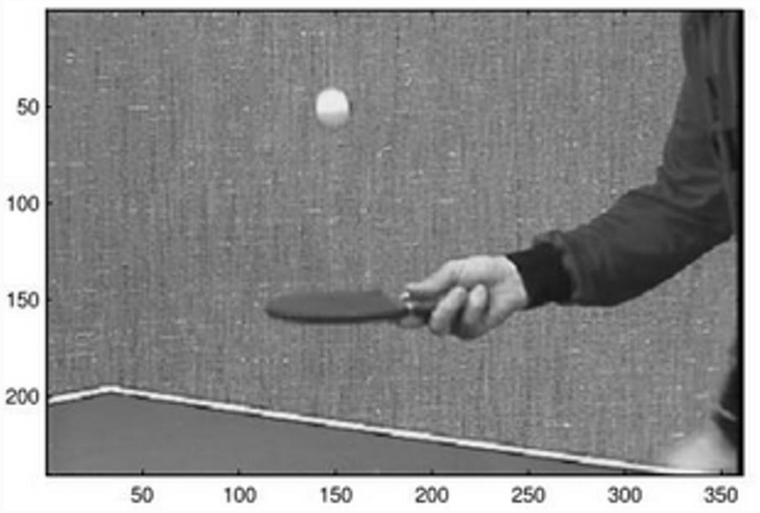


Current Frame

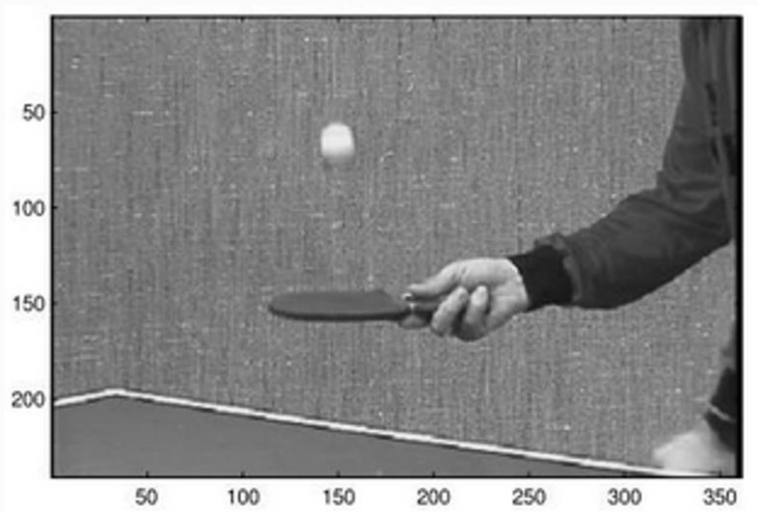


Reference Frame

Frame Difference



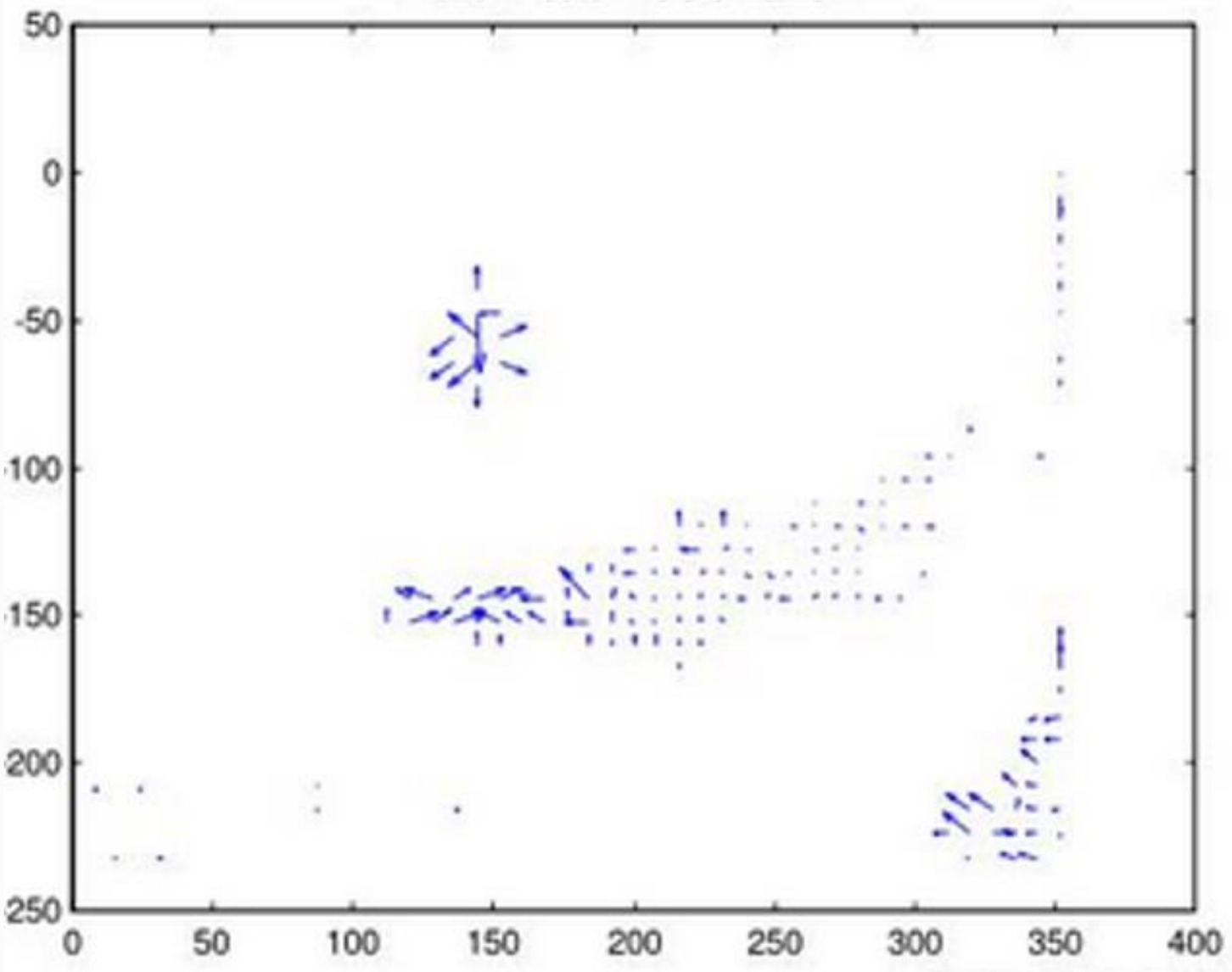
Frame 1



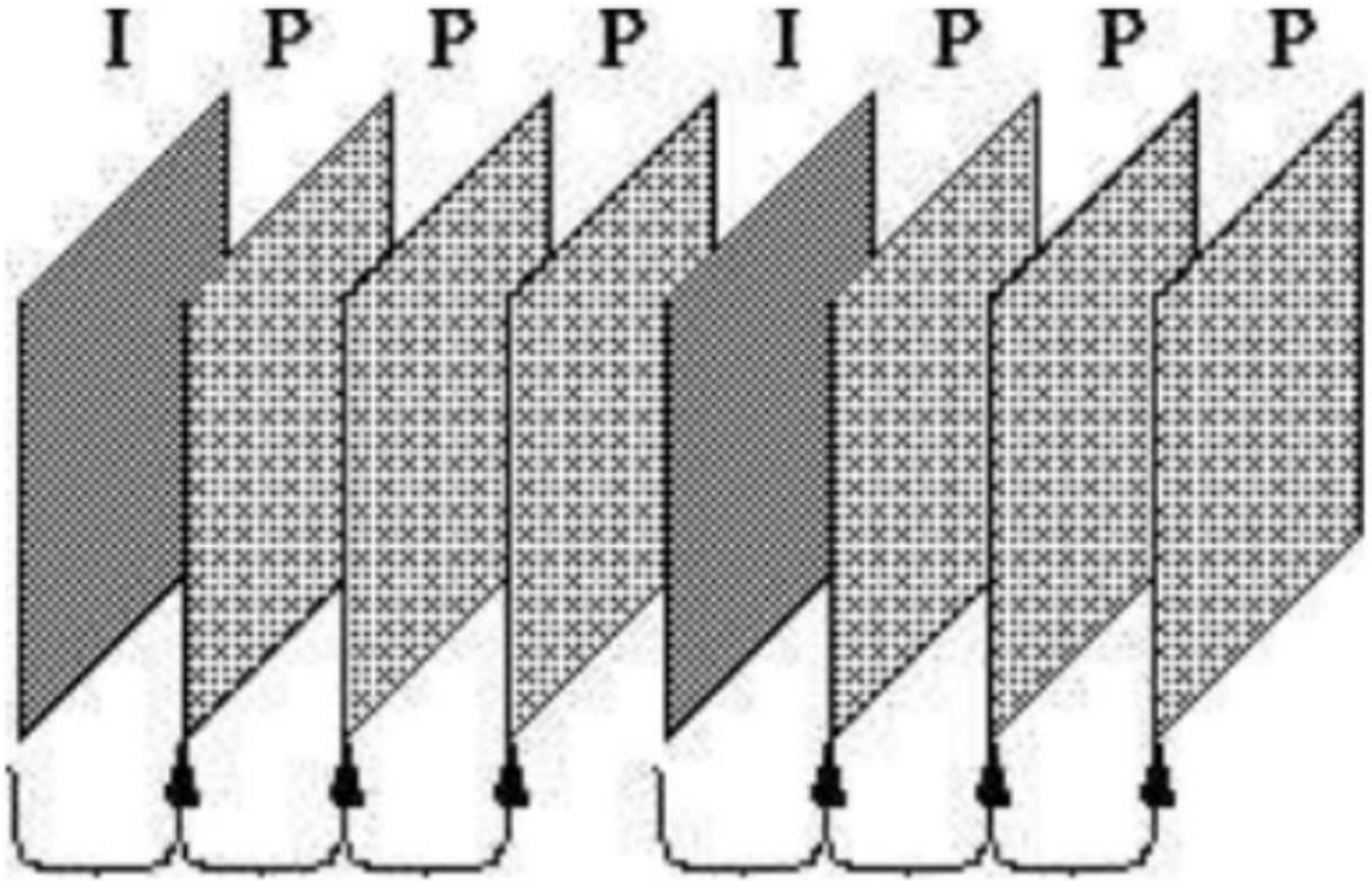
Frame 2

Frame Difference

Motion Estimation



Frame Types



Project

- فیلم نمونه ارائه شده را به سطح خاکستری (Gray Level) تبدیل کنید.
- نمونه coder و decoder ارائه شده را پیاده سازی کنید.
- Motion Estimation را به Coder و Decoder اضافه کنید.
- به ترتیب یک فریم به صورت i و سپس سه فریم به صورت p ذخیره کنید، این کار را تا پایان ویدیو انجام دهید.
- برنامه ارائه شده باید قادر به تولید یک فایل از ویدیو کد شده (در Coder) و تبدیل آن به ویدیو (در Decoder) را داشته باشد.
- فایل میانی تولید شده باید حجمی کمتر از فایل اولیه داشته باشد.
- ذخیره سازی ویدیو به صورت رنگی (RGB) نمره اضافه خواهد داشت.
- موعد تحویل ۳ تیر خواهد بود.


Q
A