Video/Image Compression Technologies

An Overview

Ahmad Ansari, Ph.D., Principal Member of Technical Staff
SBC Technology Resources, Inc.
9505 Arboretum Blvd.
Austin, TX 78759
(512) 372 - 5653
aansari@tri.sbc.com

Video Compression, An Overview

- *Introduction*
  - *Impact of Digitization, Sampling and Quantization on Compression*
- *Lossless Compression*
  - *Bit Plane Coding*
  - *Predictive Coding*
- *Lossy Compression*
  - *Transform Coding (MPEG-X)*
  - *Vector Quantization (VQ)*
  - *Subband Coding (Wavelets)*
  - *Fractals*
  - *Model-Based Coding*

Introduction

- ***Digitization Impact***
  - ***Generating Large number of bits; impacts storage and transmission***
    - » ***Image/video is correlated***
    - » ***Human Visual System has limitations***

- ***Types of Redundancies***
  - ***Spatial - Correlation between neighboring pixel values***
  - ***Spectral - Correlation between different color planes or spectral bands***
  - ***Temporal - Correlation between different frames in a video sequence***

- ***Know Facts***
  - ***Sampling***
    - » ***Higher sampling rate results in higher pixel-to-pixel correlation***
  - ***Quantization***
    - » ***Increasing the number of quantization levels reduces pixel-to-pixel correlation***

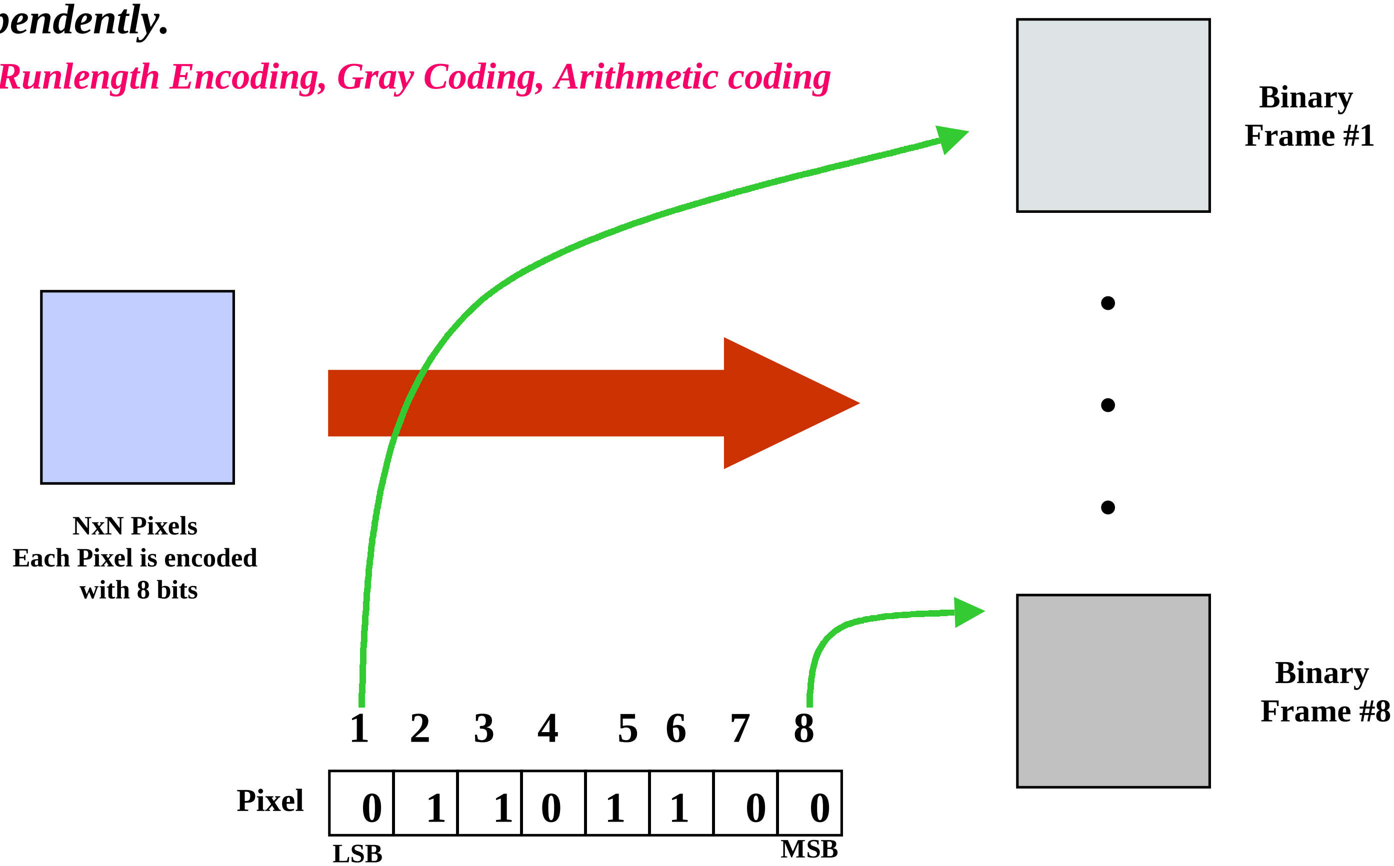
Lossless Compression

Lossless Compression

- *Lossless*
  - *Numerically identical to the original content on a pixel-by-pixel basis*
  - *Motion Compensation is not used*
- *Applications*
  - *Medical Imaging*
  - *Contribution video applications*
- *Techniques*
  - *Bit Plane Coding*
  - *Lossless Predictive Coding*
    - *DPCM, Huffman Coding of Differential Frames, Arithmetic Coding of Differential Frames*

Lossless Compression

- *Bit Plane Coding*
  - *A video frame with NxN pixels and each pixel is encoded by "K" bits*
  - *Converts this frame into K x (NxN) binary frames and encode each binary frame independently.*
    - » *Runlength Encoding, Gray Coding, Arithmetic coding*



Lossless Compression

- ***Lossless Predictive Coding***
  - ***DPCM: The goal is to find:***

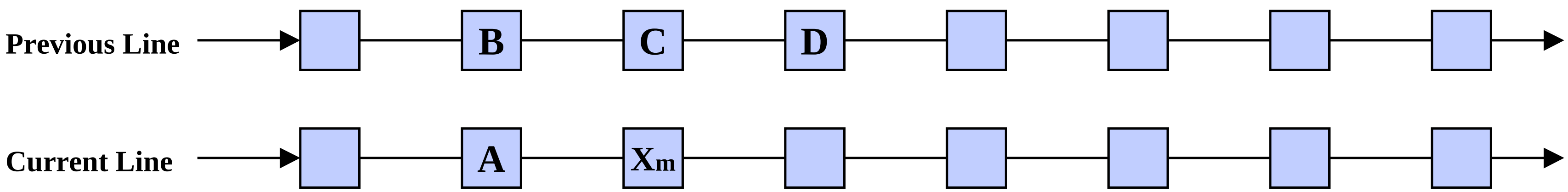
$$X'_m \quad \text{an estimate of } X_m \text{ that maximizes the following propability:}$$

$$p(X_m \mid X_{m-1,} \cdots, X_0)$$

  - ***The difference between the actual pixel value, and its most likely prediction is called the differential or the error signal:***
  - $$e_m = X_m - X'_m$$
  - ***The value of this error signal is usually entropy coded.***

Lossless Compression

■ *Lossless Predictive Coding*



$$X_m^{'} = 0.75A - 0.50B + 0.75C$$

The number of pixels used in the predictor, m, is called the predictor order and has a direct impact on the predictor's performance

- **Global Prediction:** **Same predictor coefficients for all frames**
- **Local Prediction:** **Coefficients vary from frame to frame**
- **Adaptive Prediction:** **Coefficients vary within each frames**

Lossy Compression

Lossy Compression

- ## *Transform Coding*
  - ***Desirable characteristics:***
    - » ***Content decorrelation: packing the most amount of energy in the fewest number of coefficients***
    - » ***Content-Independent basis functions***
    - » ***Fast implementation***
  - ***Available transformations:***
    - » ***Karhunen-Loeve Transform (KLT)***
      - Basis functions are content-dependent
      - Computationally complex
    - » ***Discrete Fourier Transform (DFT/FFT)***
      - Real and Imaginary components (Amplitude and Phase)
      - Fast Algorithms
    - » ***Discrete Cosine Transform (DCT)***
      - Real transformation
      - Fast algorithm
      - Best energy packing property
    - » ***Walsh-Hadamard Transform (WHT)***
      - Poor energy packing property
      - Simple hardware implementation, low-cost and fast

DCT Compression

■ *Transform Coding*

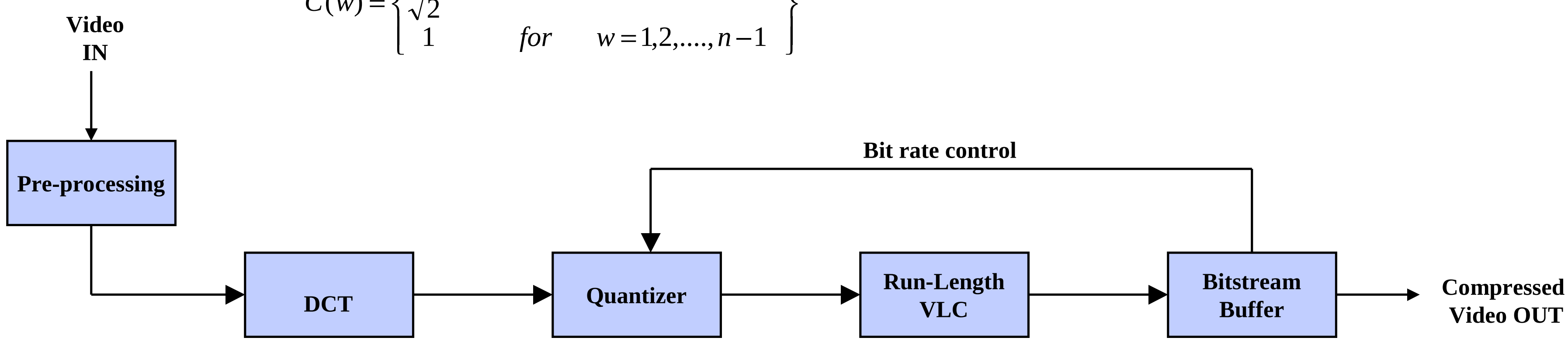
– ***DCT***

$$F(u,v) = \frac{4C(u)C(v)}{n^2} \sum_{j=0}^{n-1} \sum_{k=0}^{n-1} f(j,k) \cos\left[\frac{(2j+1)u\pi}{2n}\right] \cos\left[\frac{(2k+1)v\pi}{2n}\right]$$

$$f(j,k) = \sum_{u=0}^{n-1} \sum_{v=0}^{n-1} C(u)C(v)F(u,v) \cos\left[\frac{(2j+1)u\pi}{2n}\right] \cos\left[\frac{(2k+1)v\pi}{2n}\right]$$

where

$$C(w) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for} \quad w = 0 \\ 1 & \text{for} \quad w = 1,2,\ldots,n-1 \end{cases}$$



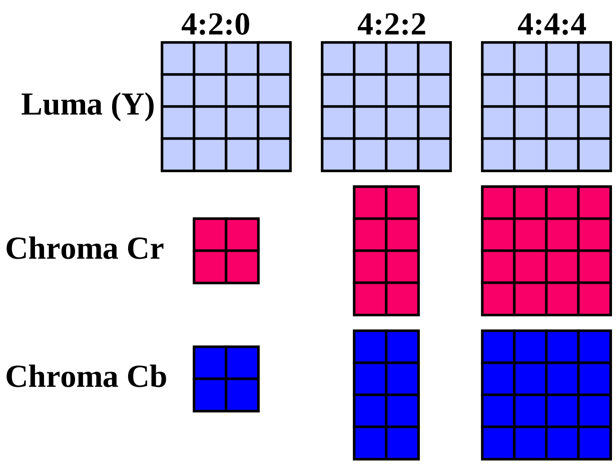
DCT Compression

Transform Coding - DCT

- *JPEG/MPEG (ISO/IEC) as well as ITU H.26x standards are based on DCT*
- *MPEG-1 was Finalized in 1991*
  - *Video Resolution: 352 x 240 (NTSC) - 352 x 288 (PAL)*
  - *30 Frames/second - Optimized for 1.5 Mbps*
  - *Supports higher bit rate and up to 4095 x 4095 resolutions*
  - *Progressive frames only*
- *MPEG-2 Finalized in 1994*
  - *Field-interlaced video*
  - *Levels and profiles*
    - Profiles: Define bit stream scalability and color space resolutions
    - Levels: Define image resolutions and maximum bit-rate per profile

MPEG-2 Compression

- ***Transform Coding - MPEG-2***
  - ***Chrominance Subsampling and video formats***
    - » *Most of the information is in the Y plane*



Format	Y samples per Line	Y lines per frame	C samples per line	C Lines per frame	Horizontal Subsampling Factor	Vertical Subsampling Factor
4:4:4	720	480	720	480	None	None
4:2:2	720	480	360	480	2x1	None
4:2:0	720	480	360	240	2x1	2x1
4:1:1	720	480	180	480	4x1	None
4:1:0	720	480	180	120	4x1	4x1

MPEG-2 Compression

Transform Coding - MPEG-2

– *Profiles and Levels*

Levels \ Profiles	Simple (SP	MAIN	4:2:2	SNR	Spatial	High
High		4:2:0 1920 x 1152 90 Mbps 60 fps				4:2:0 or 4:2:2 1920 x 1152 100 Mbps 60 fps
High 1440		4:2:0 1440 x 1152 60 fps			4:2:0 1440 x 1152 60 Mbps 60 fps	4:2:0 or 4:2:2 1440 x 1152 80 Mbps 60fps
MAIN	4:2:0 720 x 576 15 Mbps No Bs	4:2:0 720 x 576 15 Mbps 30 fps	4:2:2 720 x 576 50 Mbps 30 fps	4:2:0 15 Mbps 720 x 576		4:2:0 or 4:2:2 720 x 576 20 Mbps
Low		4:2:0 352 x 288 4Mbps		4:2:0 352 x 288 15 Mbps		

MPEG-2 Compression

Transform Coding - MPEG-2

- *MPEG-2 Picture Types*
  - ***I frame:** an Intra-coded picture that is encoded using information from itself (JPEG-like)*
  - ***P frame:** a Predictive-coded picture which is encoded using motion compensated prediction from past reference frame or past reference field*
  - ***B frame:** a Bidirectionally predictive-coded picture which is encoded using motion compensated prediction/interpolation from a past and future reference frames*
  - ***GOP Structure: IBP…IB., IBBP…IBBP….***
    - Delay and Quality are impacted by the GOP structure
- *MPEG-3*
  - ***Originally planed for HDTV, but HDTV became an extension to MPEG-2 and MPEG-3 was abandoned***
    - HDVT: 1920 x 1080 at 30 fps (1.5 Gbps)
    - 6 MHz channel bandwidth, it will only supports 19.2 Mbps (video is only around 18Mbps)

MPEG-4 Compression

- ***MPEG-4***
  - *Like other MPEG standards is an ISO/IEC standard*
    - » *It provides technologies to view access and manipulate objects rather than pixels*
    - » *Provides tools for shape coding, motion estimation and compensation, and texture coding*
    - » *Shape encoding can be performed in binary mode or gay scale mode*
    - » *Motion compensation is block-based (16x16 or 8x8) with half pixel resolution, and provides a mode for overlapped motion compensation*
    - » *Texture encoding is done with DCT (8x8 pixel blocks) or Wavelets*

MPEG-4 Compression

- ***MPEG-4***

  – ***Profiles***

    » ***Simple Profile and Core Profile***
      - QCIF, CIF
      - Bit rates: 64 Kbps, 128 Kbps, 384 Kbps and 2 Mbps

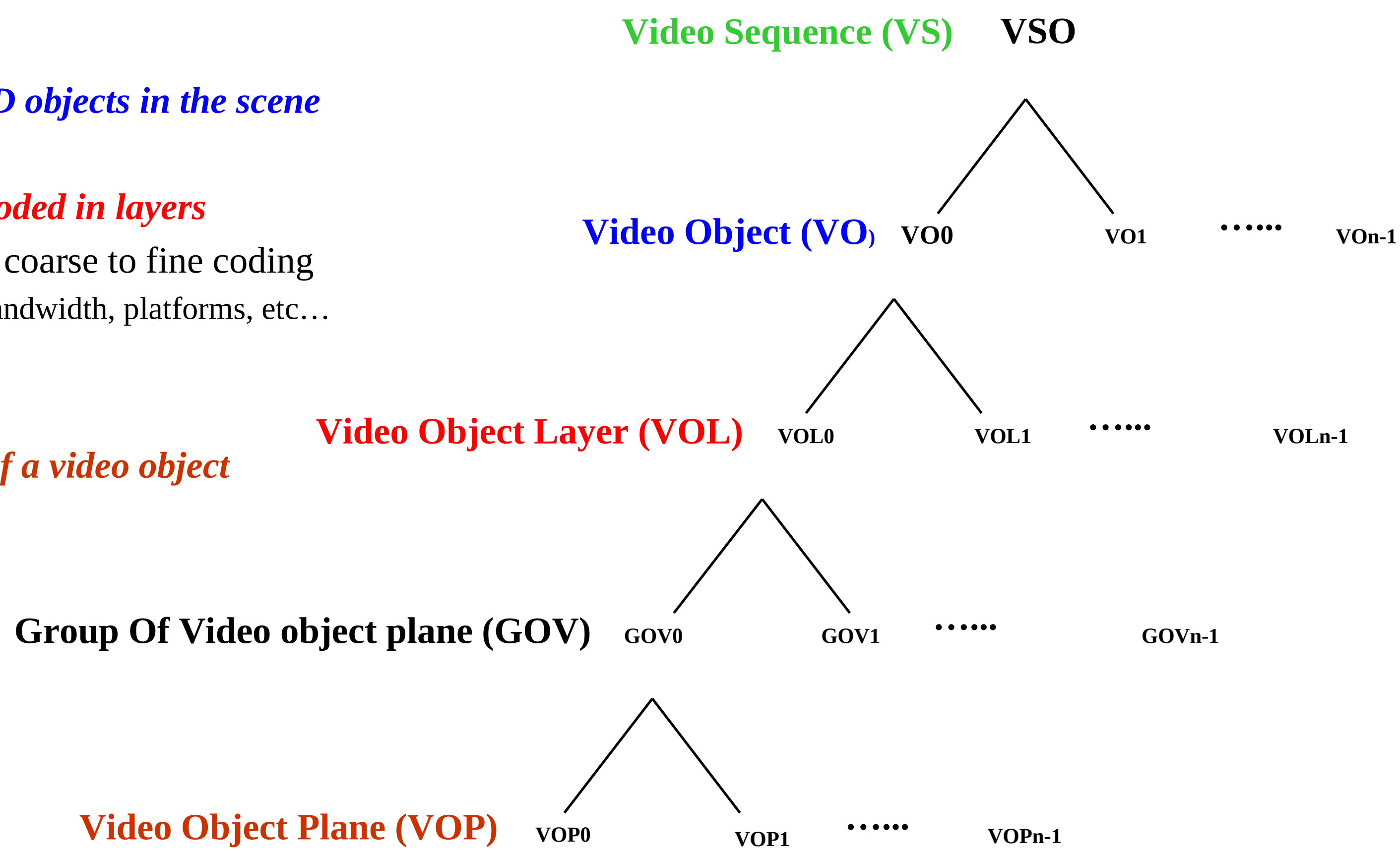
    » ***Main Profile***
      - CIF, ITU-R 601 and HD
      - Bit rate: 2 Mbps, 15 Mbps, 38.4 Mbps

    » ***MPEG-4 has been explicitly optimized for three bit-rate ranges:***
      - Below 64 Kbps
      - 64 - 384 Kbps
      - 384 - 4 Mbps

    » ***Chrominance format supported is 4:2:0***

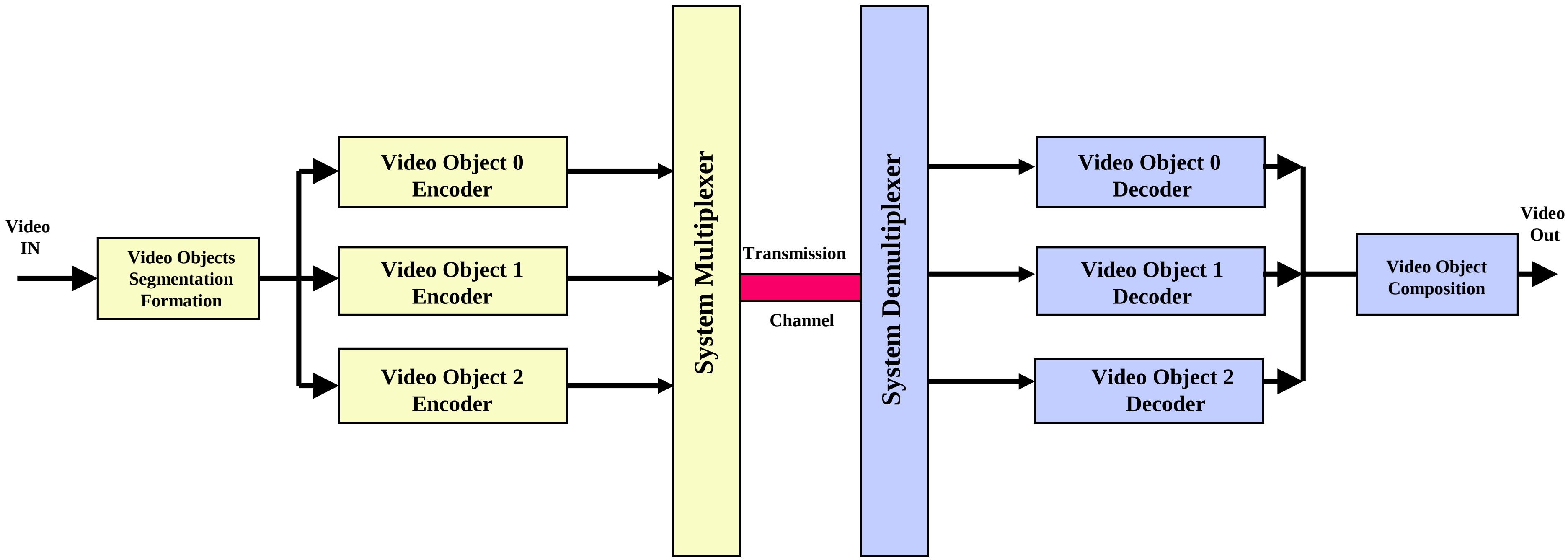
MPEG-4 Compression

- ***Data Structure in visual part of MPEG-4***
  - » *VS: A complete Scene*
  - » *VO: Video objects; 2-D objects in the scene*
  - » *VOL: Each VO is encoded in layers*
    - Scalability from coarse to fine coding
      - Multiple Bandwidth, platforms, etc…
  - » *GOV (Optional)*
  - » *VOP: A time sample of a video object*



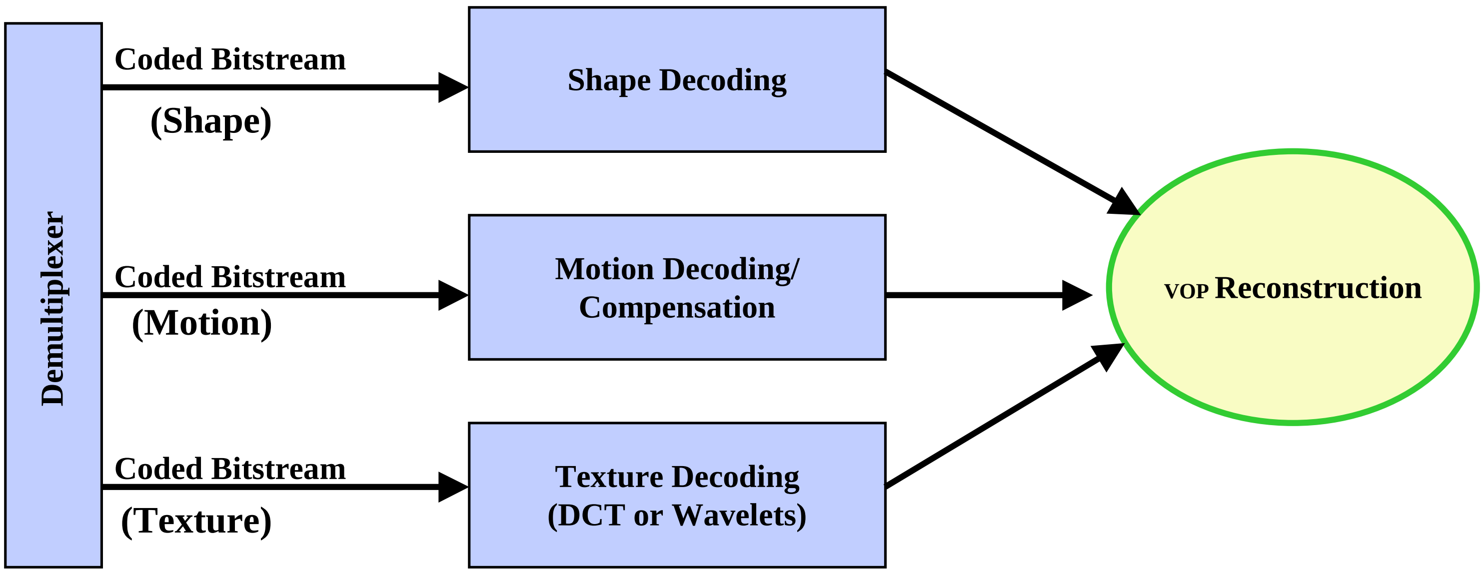
MPEG-4 Compression

- ## *MPEG-4*
  - *General Block Diagram*



MPEG-4 Compression

- ***MPEG-4***
  - *Video Object Decoding*



MPEG-4 Compression

- *MPEG-4*
  - *Coding Tools*
    - » *Shape coding: Binary or Gray Scale*
    - » *Motion Compensation: Similar to H.263, Overlapped mode is supported*
    - » *Texture Coding: Block-based DCT and Wavelets for Static Texture*
  - *Type of Video Object Planes (VOPs)*
    - » *I-VOP: VOP is encoded independently of any other VOPs*
    - » *P-VOP: Predicted VOP using another previous VOP and motion compensation*
    - » *B-VOP: Bidirectional Interpolated VOP using other I-VOPs or P-VOPs*
    - » *Similar concept to MPEG-2*

MPEG-7: Content Description

- ***MPEG-7 (work started in 1998)***
  - ***A content description standard***
    - » *Video/images: Shape, size, texture, color, movements and positions, etc…*
    - » *Audio: Key, mood, tempo, changes, position in sound space, etc…*
  - ***Applications:***
    - » *Digital Libraries*
    - » *Multimedia Directory Services*
    - » *Broadcast Media Selection*
    - » *Editing, etc…*

Example: Draw an object and be able to find object with similar characteristics.
Play a note of music and be able to find similar type of music

VQ Compression

- ## *Vector Quantization (VQ)*
  - ***Shannon's Theory***
    - » ***Purely digital signals could be compressed by assigning shorter code-words to more probable signals and that the maximum achievable compression could be determined from a statistical description of the signal***
    - » ***Coding vectors or groups of symbols (speech samples or pixels), rather than individual symbols or samples***

• Each Image vector X is compared with a collection of codevectors $Y_{i\,,\,i\,=}$ 1, 2, 3, ….., n taken from a previously generated Codebook.

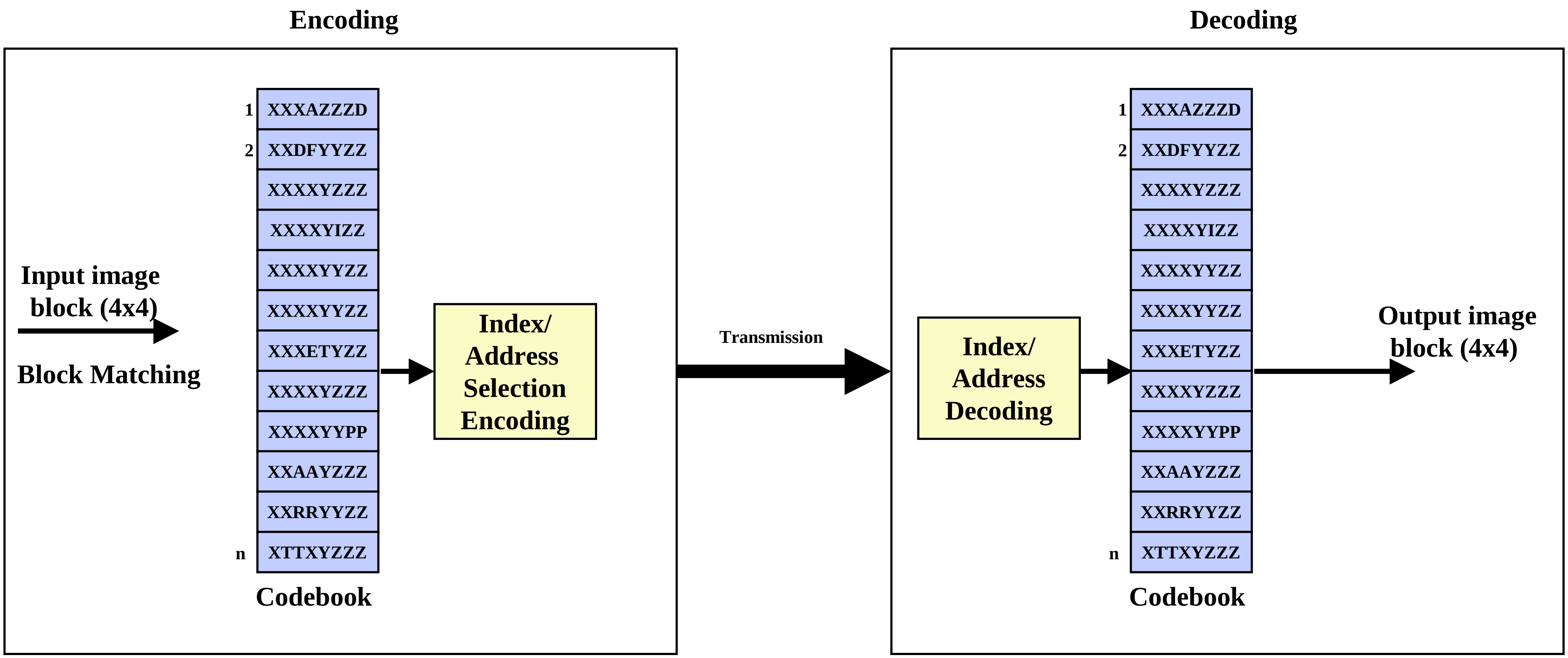
• The best match codevector is chosen using a minimum distortion rule:

$$d(X,Y_k) \quad \leq \quad d(X,Y_j) \quad \text{for all} \;\; j = 1,2,\ldots..,n$$

Where d(X,Y) denotes the distortion incurred in replacing the original X with Y. MSE is widely used because of its computational complexity.

VQ Compression

■ *Vector Quantization*



- **Each frame is divided into small blocks (4x4 pixels)**
- **A Codebook containing blocks of 4x4 pixels is designed using clustering techniques and training content**
- **The best match for each input block is found from the Codebook**
- **The index for the best matched block from the Codebook is transmitted**

VQ Compression

- ***Vector Quantization***
  - ***Codebook Generation***
    - » ***Best results are obtained when the codebook is generated from the content itself (Local codebooks)***
      - Computationally intensive task
      - Creates overhead - codebook has to be transmitted to the receiver as overhead
    - » ***Global Codebooks***
      - Linde-Buzo-Gray (LBG) clustering algorithm
        - **Training content from the same class of content is used**
        - **The larger the codebook the higher the bit rate, the higher the quality of the content**

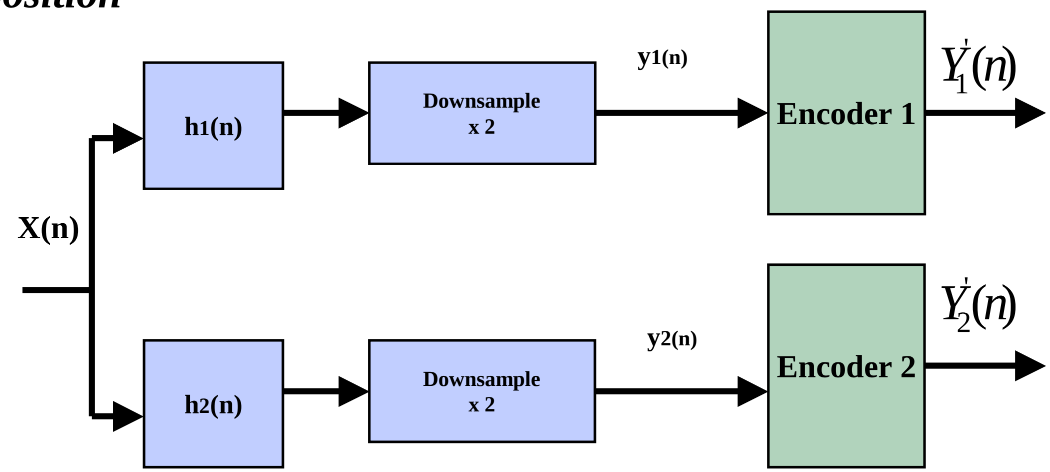
Subband Compression

- ***Subbband Coding***
  - ***Analysis Stage***
    - » ***Each frame is filtered to create a set of smaller frames (subbands)***
      - Each smaller frame contains a limited range of spatial frequencies
      - Since each subband has a reduced bandwidth compared to the original full-band frame, they may be downsampled.
    - » ***Each band is encoded separately (different bit rates, encoder, etc…)***
  - ***Synthesis Stage***
    - » ***Reconstruction the original frame from its subbands***
      - Each band is decoded and then upsampled
      - The appropriate filtering is applied on each subband
      - Subbands are added together to reconstruct the original fram

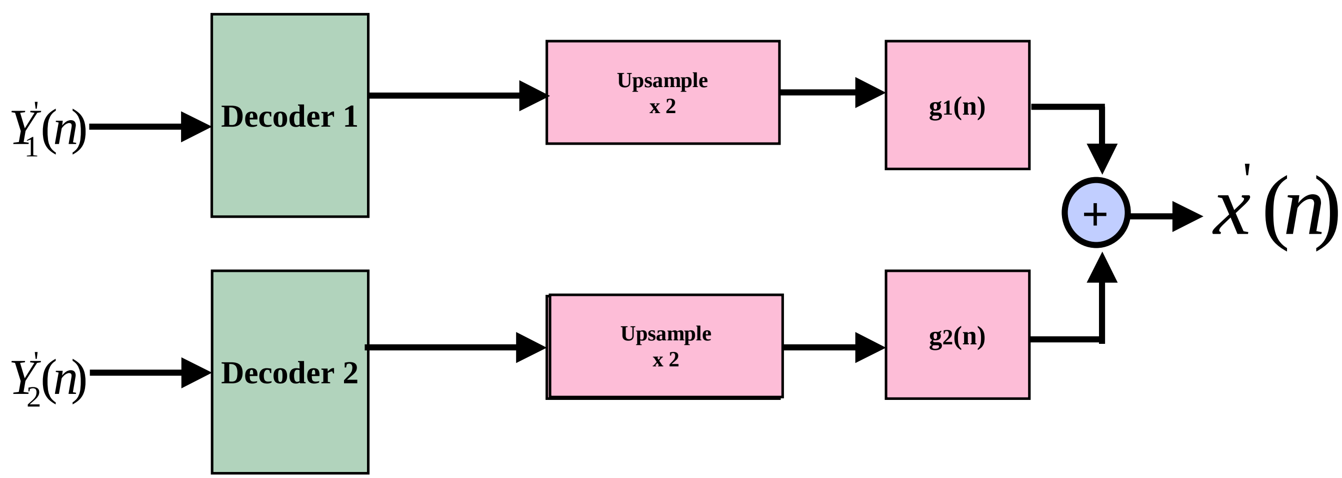
Subband Compression

- **Subbband Coding**
  - *Example of two band decomposition*

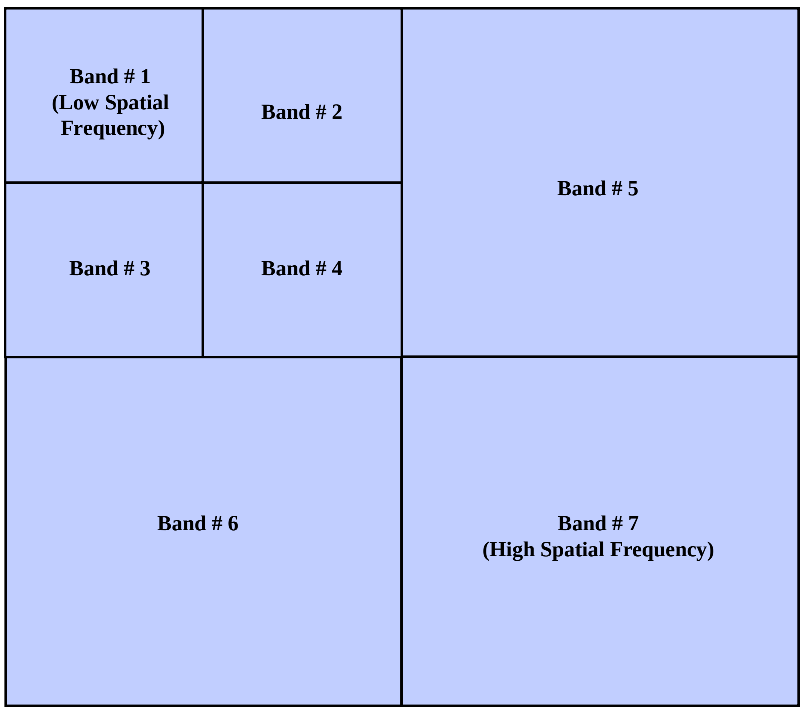
Analysis Stage

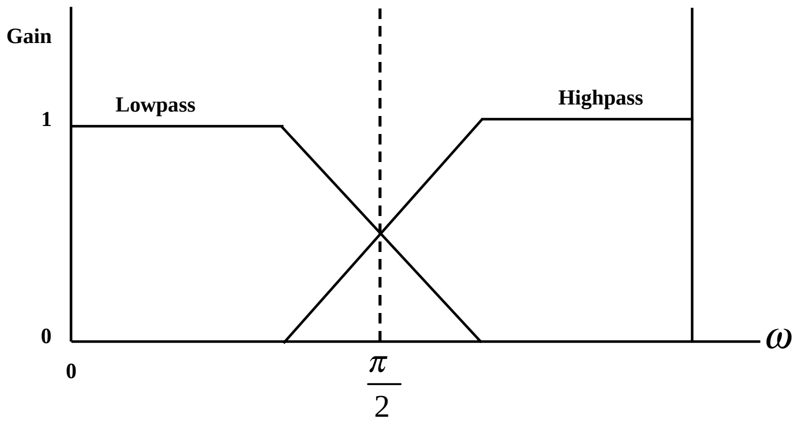


Synthesis Stage



- **Subband Coding**
  - *Analysis/Synthesis Filtering*
    - » *Quadrature Mirror Filters (QMFs)*



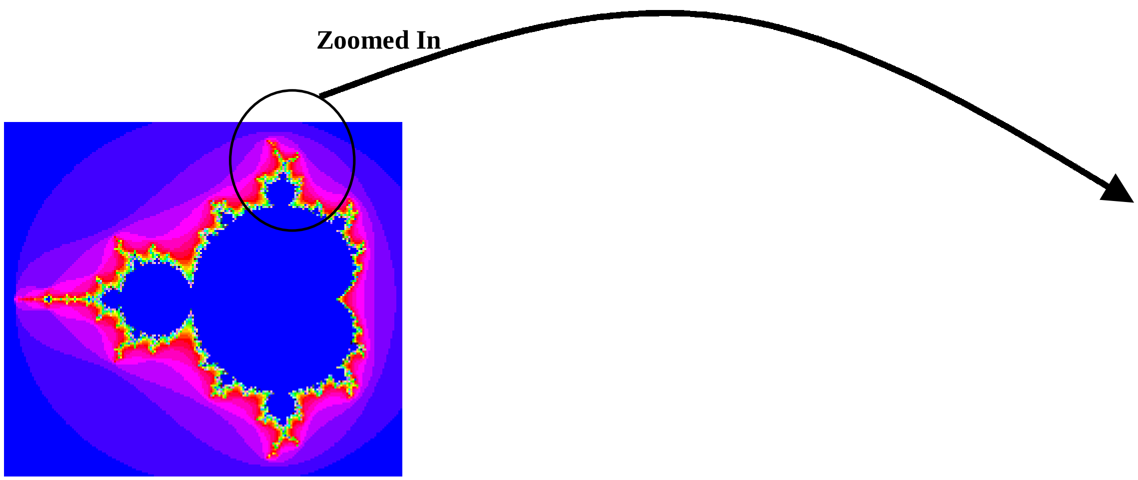


Subband Compression

- *Subbband Coding*
  - *Key Advantage*
    - » *No blocking artifacts, at lower bit rates*
    - » *Adaptive compression techniques can be applied to each subband*
  - *Key Disadvantage*
    - » *Difficult to perform motion compensation*

Fractal Compression

- ***Fractals***
  - ***Discovered by Benoit Madelbrot, a student of Gaston Julia (Julia Set) at Ecole Polytechnique de Paris***
    - » ***Fractus in Latin, is an image that is infinitely complex and self similar at different level***
    - » ***A fractal is usually a rough or broken geometric shape which can be subdivided into parts. Often they are either self-similar or quasi self-similar.***



A fractal image from Madelbrot Set

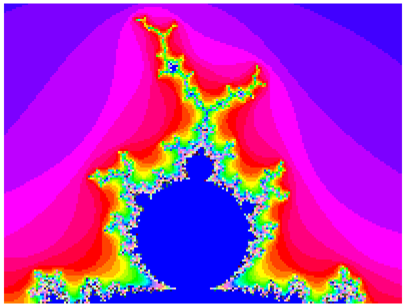


Image # 2

Fractal Compression

■ *Fractals*

Zoomed In

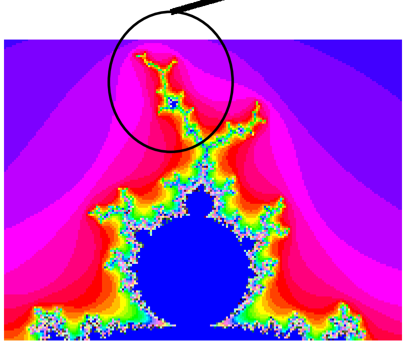


Image # 2

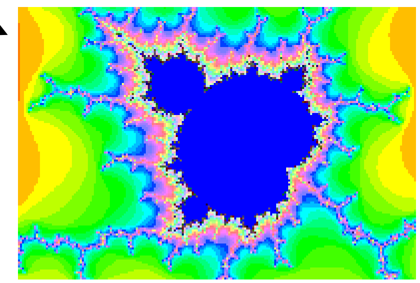
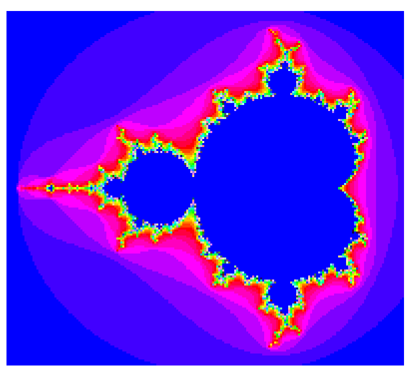


Image # 3

Image # 3 is similar to the original image with differences in color and it is rotated. This is Self similarity.



Original image

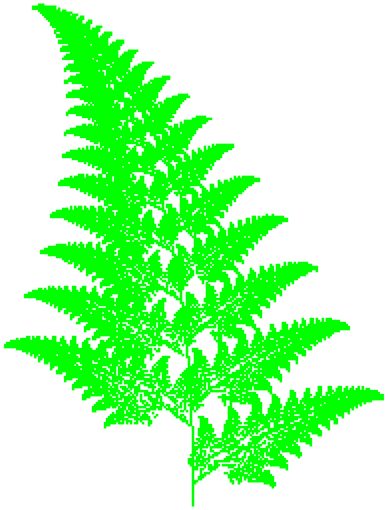
Fractal Compression

Fractals

– ***Image and Video Compression***

» ***Only 28 numbers are needed to create the image on the right. The image can be create with infinite resolution with only 28 numbers.***

0	0	0	.16	0	0	.01
.85	-.04	.04	.85	0	1.6	.85
.2	-.26	.23	.22	0	1.6	.07
-.15	.28	.26	.24	0	.44	.07



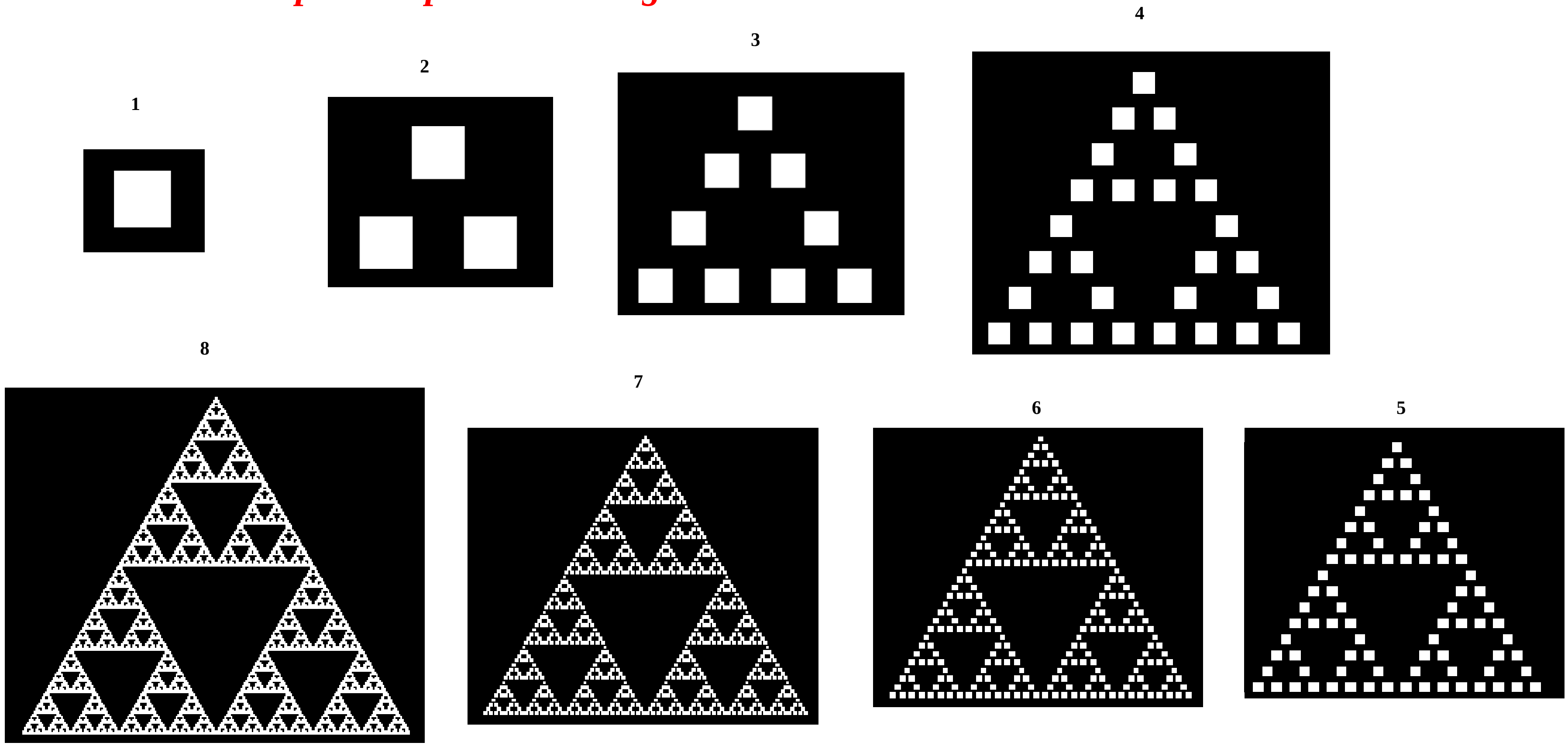
Fractal Compression

- ## *Fractals*
  - *Iterated Function Systems (IFS) and Affine Transformation*
    - » *Rotation, Movement, Size changes, are defined by an affine transform*
    - » *Example of an affine transformation with 6 constant a, b, c, d, e, f: Pixel with coordinate (x,y) is transformed as follows:*

$$X' = a * x + b * y + c$$
$$Y' = d * x + e * y + f$$

Fractal Compression

- ***Fractals***
  - *Example: Sierpinski's Triangle*



Fractal Compression

- *Fractals*
  - *Issues*
    - » *It is an inverse problem: find the affine transformation coefficients for any image; Difficult for Natural Scenes*
    - » *Extremely computationally Intensive*
    - » *Very difficult to implement in real-time*
  - *Benefits*
    - » *Very low bit rate compression*
    - » *Multi-resolution and scalable compression technique*

Model-Based Compression

- *Model-Based compression*
  - *Design models for different objects in natural scenes*
  - *Both the encoder and the decoder have copies of the models*
  - *Track changes (movement, size change, rotation) and send only those information to the decoder to change the models appropriately*
  - *Difficult to find models for all natural scenes*
  - *Computationally complex*
  - *Look promising for head and shoulder type of applications*
  - *Very low bit-rate can be achieved*

Object-Based Compression

- *Object-Based compression*
  - *Object segmentation*
  - *Motion compensation/Object Tracking*
  - *Texture encoding*

- *Issues*
  - *Computationally intensive*