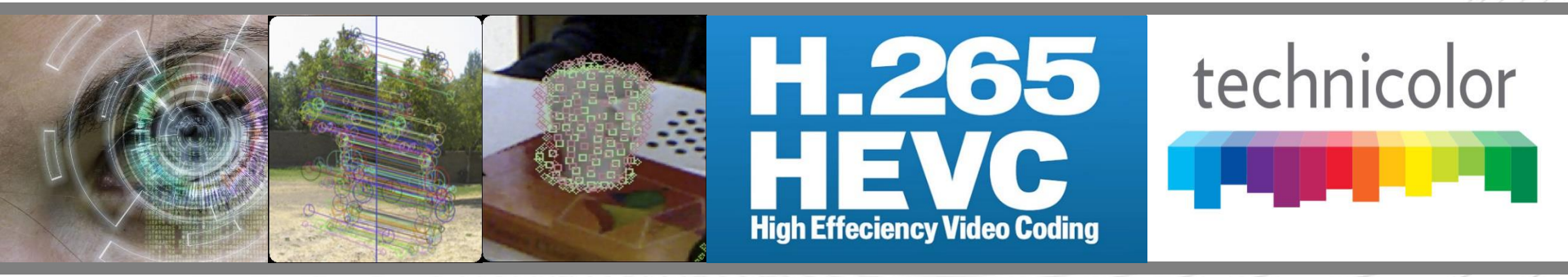
VICL
Visual Information Computing Lab

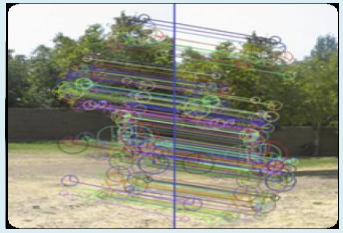
Advanced Multimedia Signal Processing (#2: Concept of Image/Video Data)

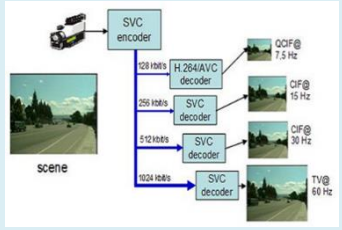


2017. Spring

Prof. Byung-Gyu Kim

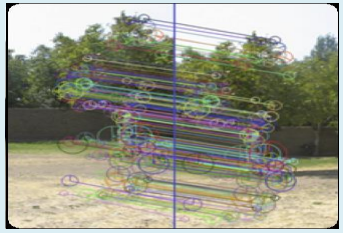
Dept. of IT Engineering, Sookmyung Women's University

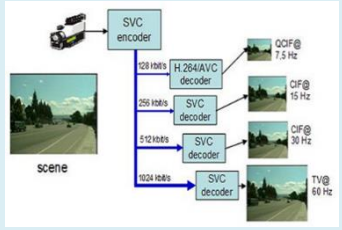




Contents

- Scheme of Multimedia Computing
- Concept of Video
- Frequency domain analysis
- Basic Processes for Video Compression





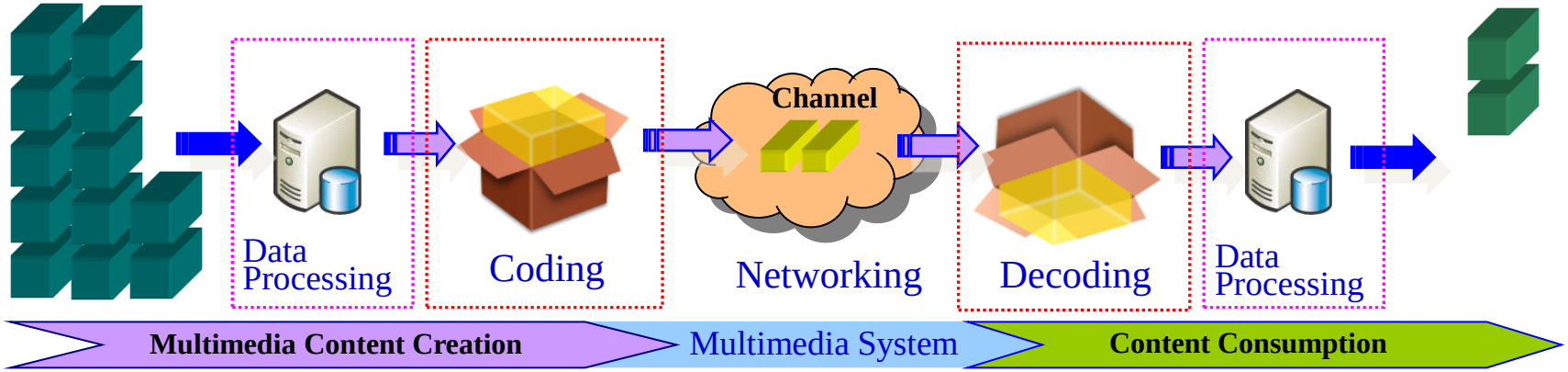
Contents

- **Scheme of Multimedia Computing**
- **Concept of Video**
- **Frequency domain analysis**
- **Basic Processes for Video Compression**

Scheme of Multimedia Computing (1)

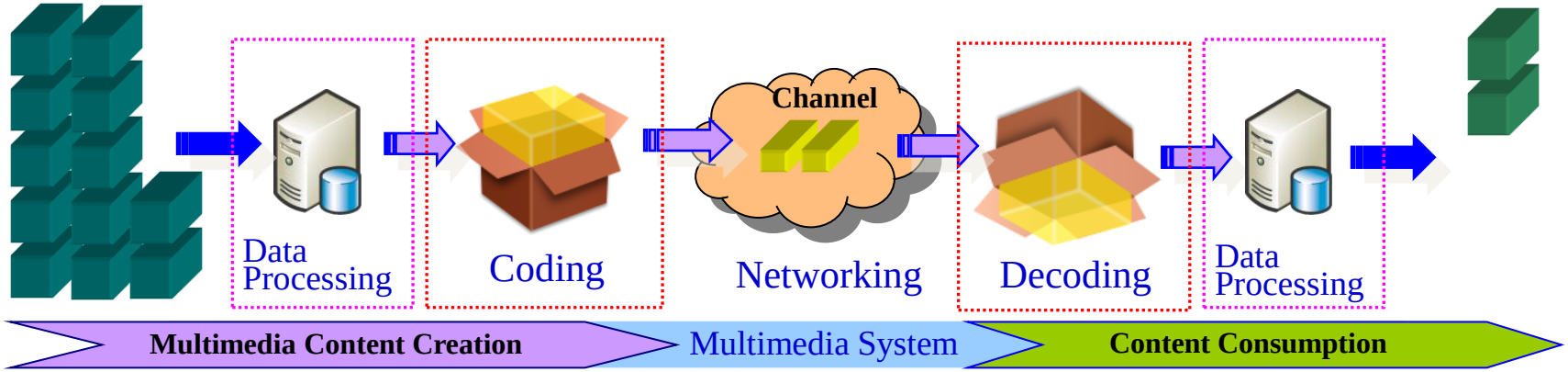
❖ Multimedia Application system

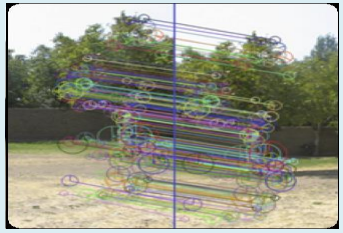
- Data Processing
  - Feature extraction, detection, and image filtering.
- Coding (decoding)
  - Compression algorithms for audio, images, and video using data correlation. MPEG audio/visual standards, H.26x series.

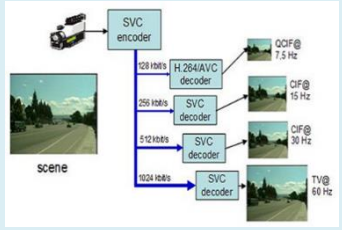


Scheme of Multimedia Computing (2)

- System
  - Integrating audio and video (muxing/demuxing) and other components. MPEG-1, 2, and 4 system for storage, MPEG2-TS for transmission.
- Networking
  - Transmission of multimedia data over networks (channels). H.32x, SIP, RTSP for IP network and IP convergence network.







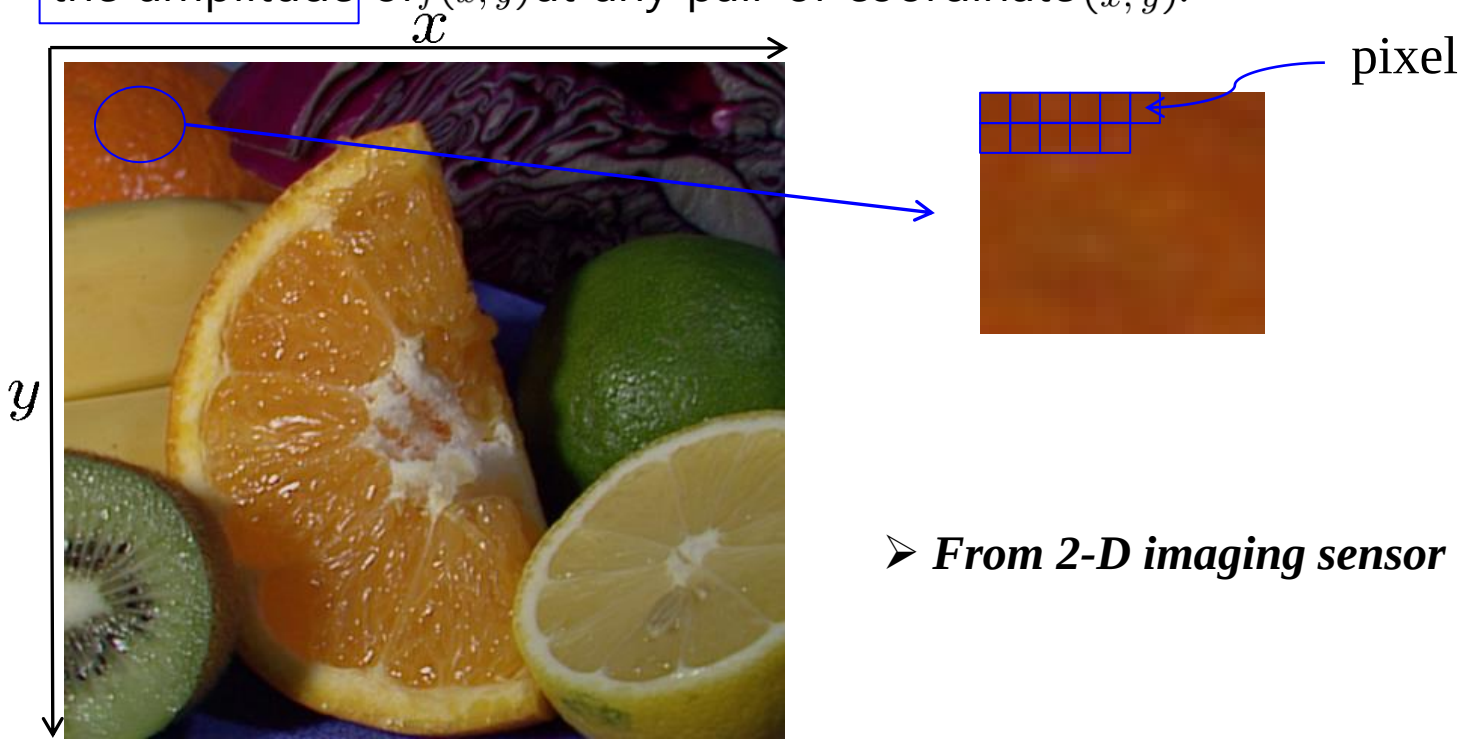
Contents

- Scheme of Multimedia Computing
- Concept of Video
- Frequency domain analysis
- Basic Processes for Video Compression

Concept of Video (0)

❖ What is a ***Digital Image***?

- May be defined as a two-dimensional function, $f(x, y)$.
  - x and y are spatial (plane) coordinates (at fixed time).
  - the amplitude of $f(x, y)$ at any pair of coordinate (x, y).

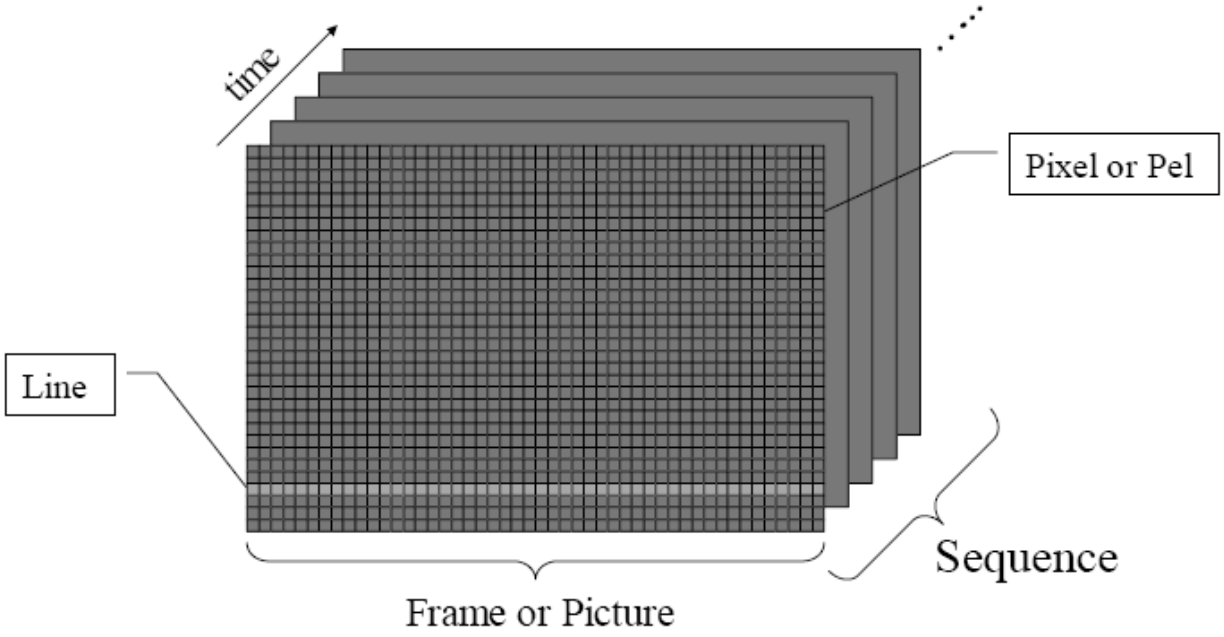


➢ ***From 2-D imaging sensor***

Concept of Video (1)

❖ What is Video (sequence)?

- Time axis (temporal axis)
- Group of the consecutive captured pictures (images) for the given time duration.



Concept of Video (2)

- Amount of Data for Video (or Image)
  - Image case:
    - 1 pixel: 8bits (256 levels)
    - Gray scale: Height x Width x Bits
    - Color: 3 bands,

      Height x Width x Bits x 3
    - RGB case:

      512 * 512 * 8 * 3

      = 6291456 bits

      = 6144 K bits

      = 6 M bits

Fig. 1. “Lena” (512x512 Still image)

How much BW we need to send this image per 1 second?

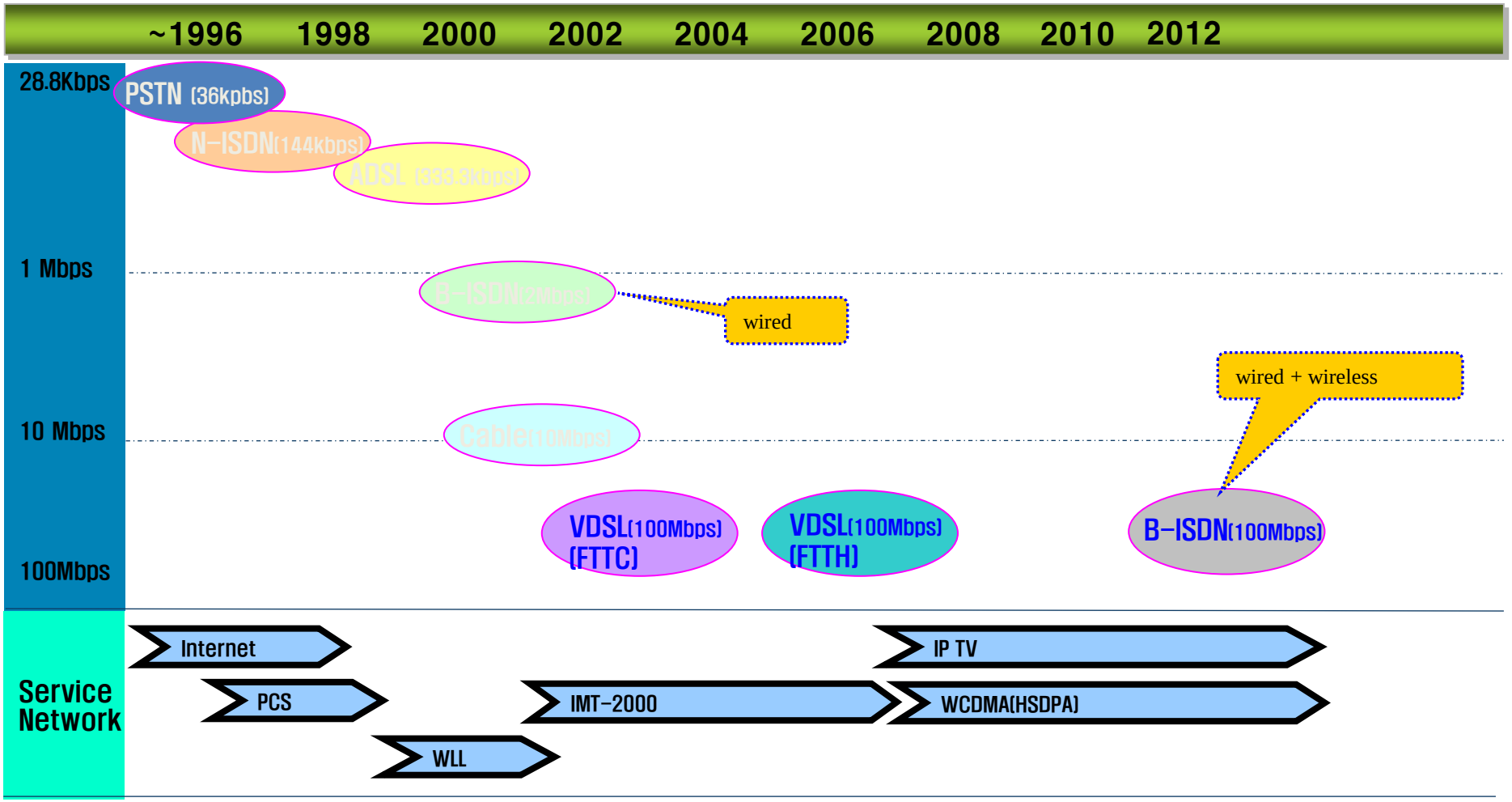
Concept of Video (3)

- Amount of Data for Video (or Image)
  - Video case:
    - So many number of this kinds of image as successive form.
    - To provide real-time media service, at least 24 frames per second should be guaranteed. Then

6 M bits x 24 = 144M bits
(if 512 x 512 size image is given)

Concept of Video (4)

❖ Evolution of Digital Network Infra



Concept of Video (5)

❖ Relationship Between Successive Frames

t-1

t

➡ Much **Correlation** between frames

Concept of Video (6)

❖ **How to use this correlation** to reduce video data?

- Spatial correlation
  - Spatial redundancy
    - Very similar intensity to its neighbor pixels spatially.

Concept of Video (7)

- Temporal correlation
  - Temporal redundancy
    - High similarity between one pixel (frame **t**)and corresponding pixel at the same position of the previous frame (frame (*t*-1)).

Pixel (x,y)

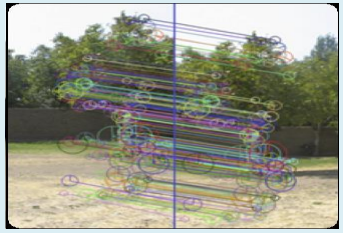
Frame 9

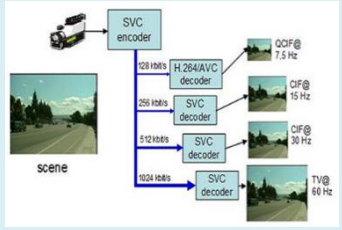
Pixel (x,y)

Pixel (x,y)

Frame 10

Difference image



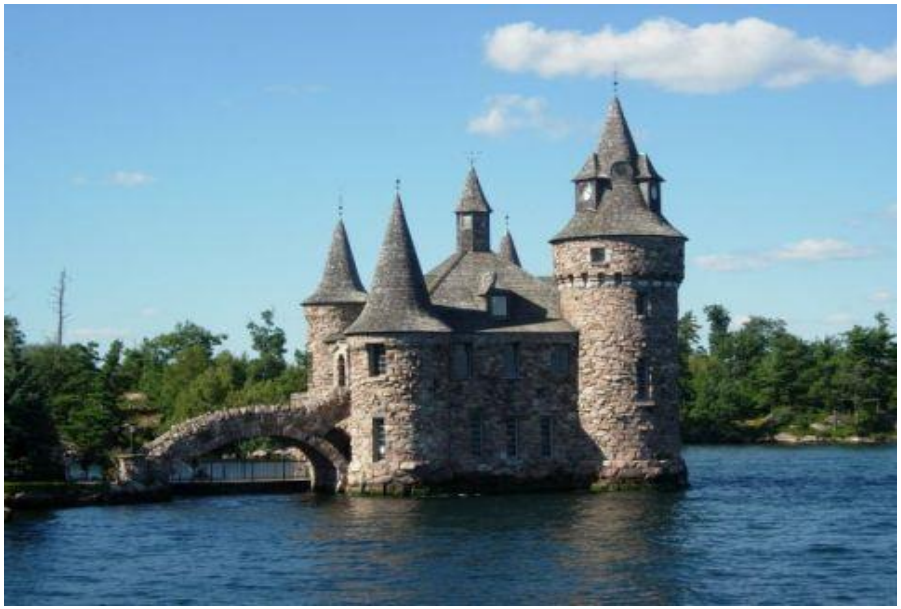


Contents

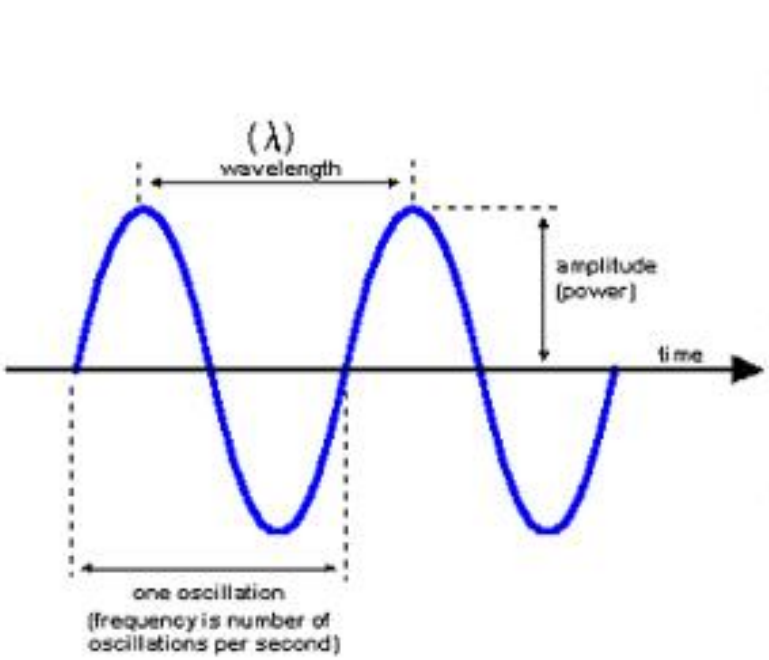
- Scheme of Multimedia Computing
- Concept of Video
- Frequency domain analysis
- Basic Processes for Video Compression

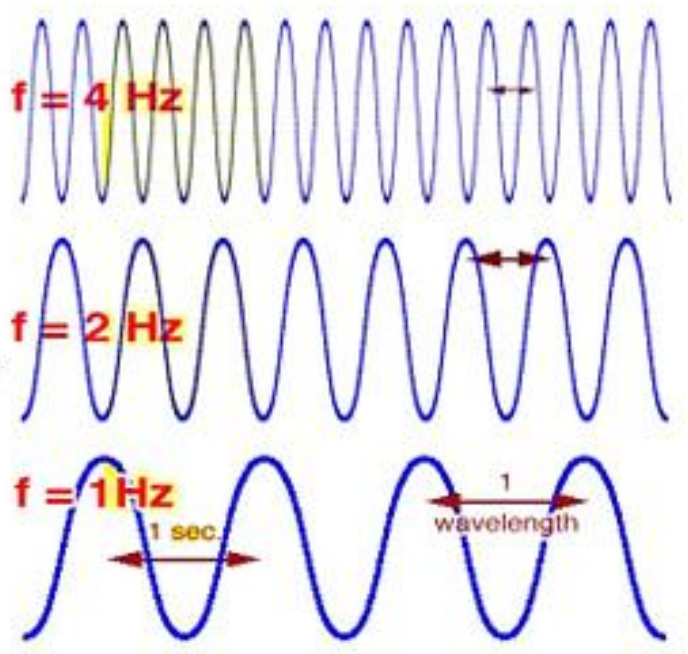
Frequency domain analysis(주파수 영역분석)

- Frequency component: A degree of signal change in time duration
- Frequency in Image domain
  - Signal = Intensity or luminance
    - Rapid variation
    - Slow variation



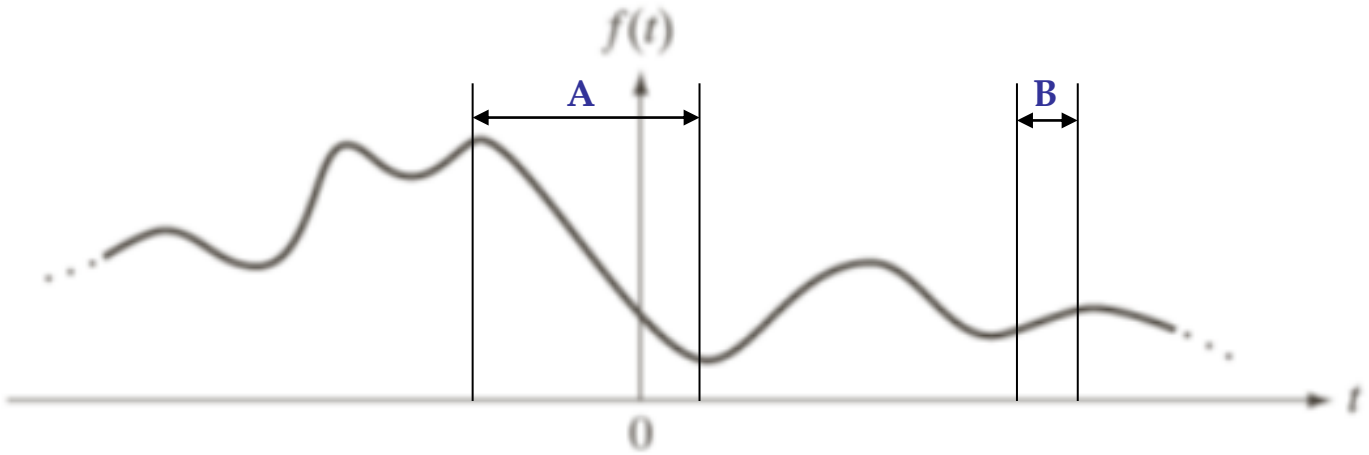
- How to define the high or low frequency component?
  - Periodic signal case
    - How many repetitions within some time duration.
    - Reverse of time period(T)





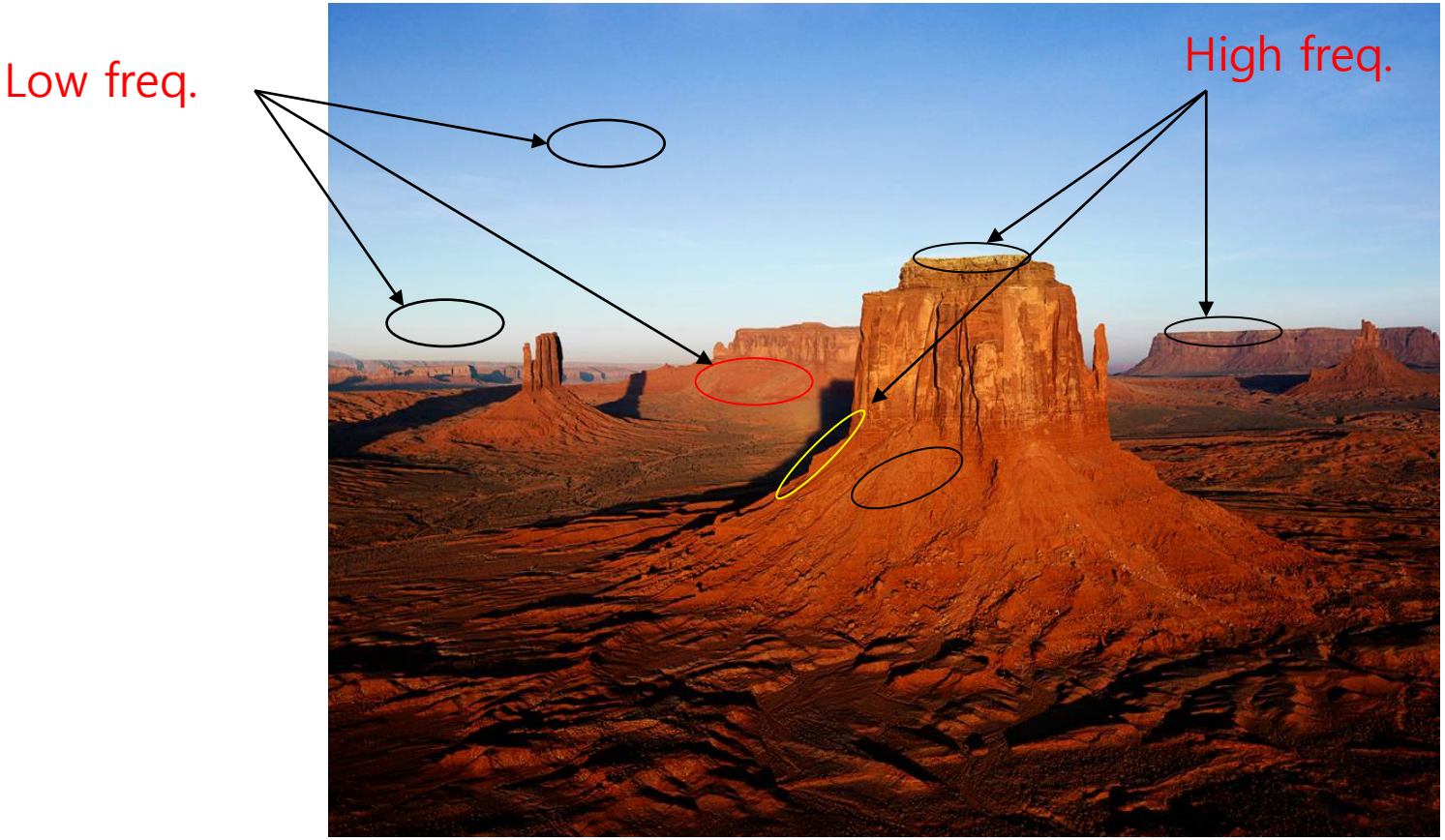
Frequency domain analysis(주파수 영역분석)

- Aperiodic signal case
  - How much variation within the given time.
    - Large change → High frequency component is dominant
    - Almost slow change → Low frequency component is dominant



Frequency domain analysis(주파수 영역분석)

- In the Image Case
  - How much variation within the given spatial domain.
    - Large change → High frequency component is dominant
    - Almost slow change → Low frequency component is dominant



Frequency domain analysis(주파수 영역분석)

- Frequency domain analysis(주파수 영역분석)
  - Representation of low freq. ~ high freq. components
  - Usual Transform To Frequency domain
    - Discrete Fourier Transform
    - Discrete Cosine Transform
    - Discrete Sine Transform 등이 있음

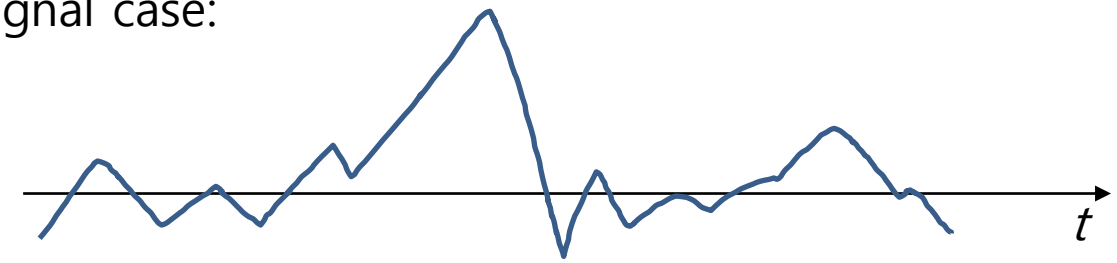
- High Pass Filter(고주파 통과 필터)
  - High freq. passed and low freq. components are removed.

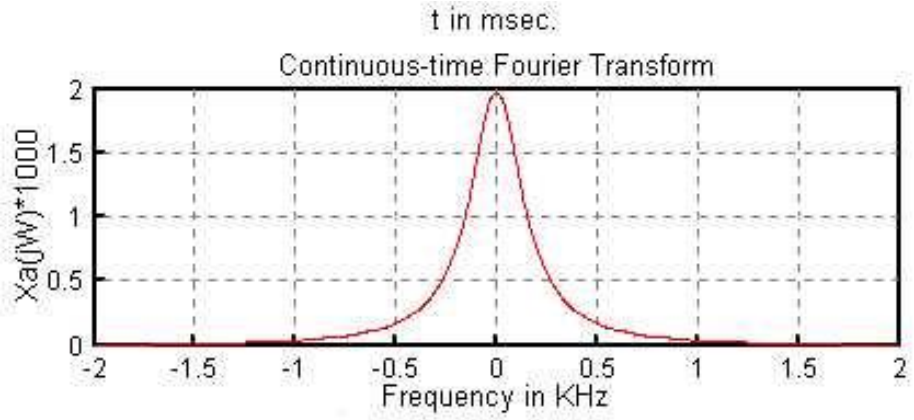
- Low Pass Filter(저주파 통과 필터)
  - Low freq. passed and high freq. components are removed.

❖ Property of Signal in Frequency Domain

- **Low frequency components are very dominant in natural signal**.

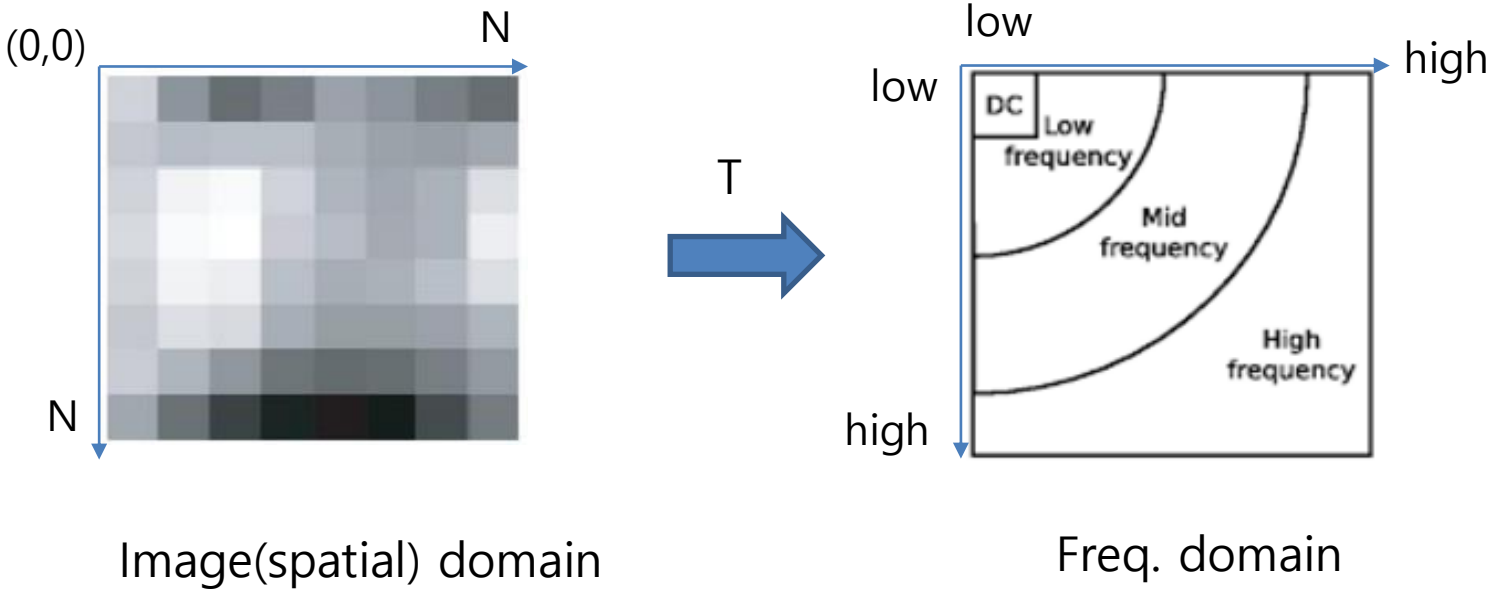
- 1-Dim. Signal case:





Frequency domain analysis(주파수 영역분석)

- 2-Dim. Signal case (8x8 block image0:
  - NxN image → NxN transform cofficients

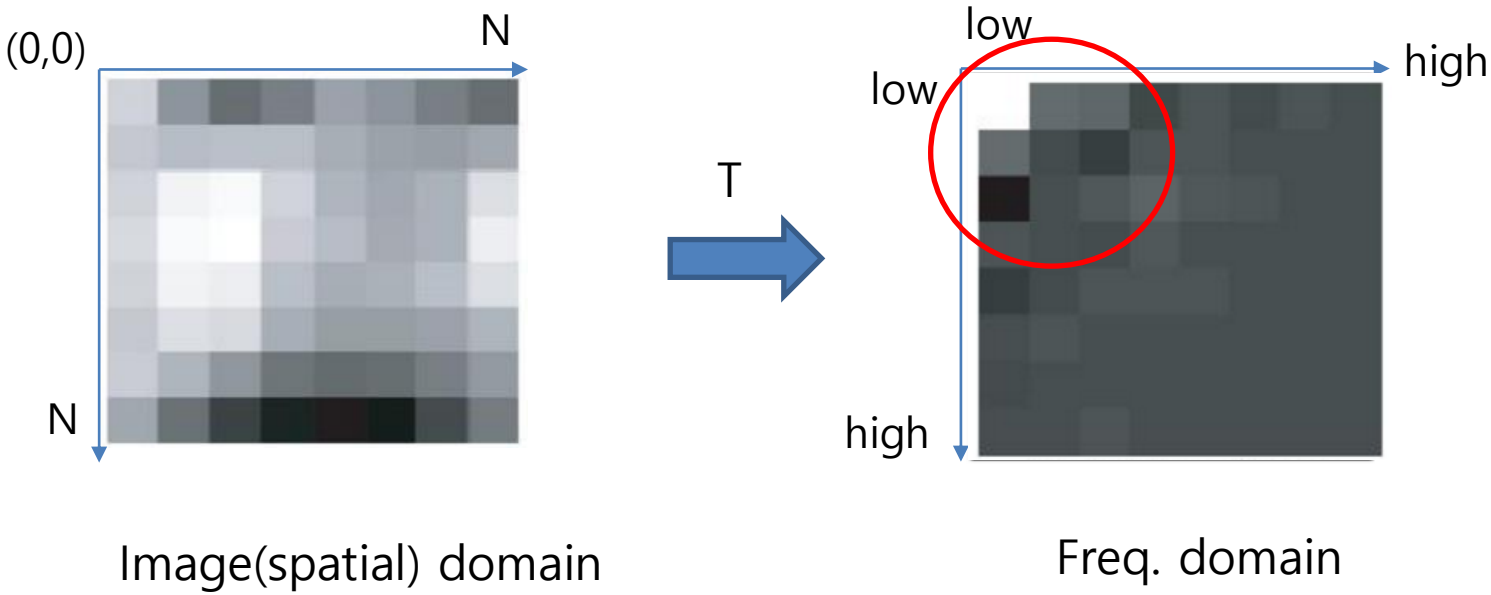


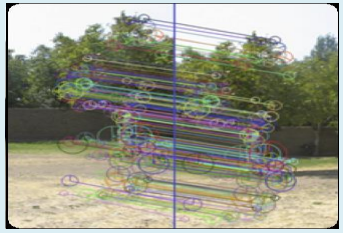
Image(spatial) domain

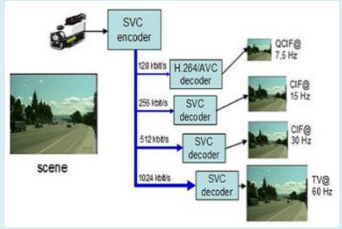
Freq. domain

Frequency domain analysis(주파수 영역분석)

- 2-Dim. Signal case (8x8 block image0:
  - NxN image → NxN transform cofficients



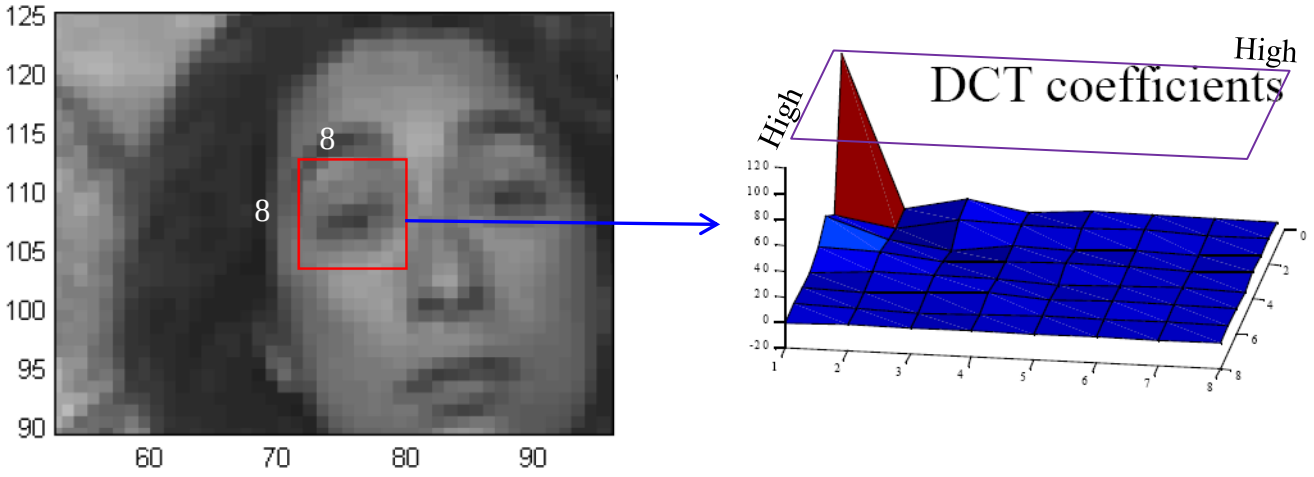




Contents

- Scheme of Multimedia Computing
- Concept of Video
- Frequency domain analysis
- Basic Processes for Video Compression

- Theoretic Tools
  - For Spatial Redundancy.
    - *Signal analysis in frequency domain.*
      - **Discrete Cosine Transform (DCT)**
      - Discrete Sine Transform (DST)
      - Discrete Fourier Transform (DFT)
      - Etc.
    - *Signal Property in frequency domain.*



Basic Processes for Video Compression

- For Temporal Redundancy (between frames),
  - *Difference image* between consecutive frames.

Residual image

  - Pixel by pixel difference

Still much data exist..!!!
Is there any way to more reduction?

Basic Processes for Video Compression

- Block-based Motion Estimation (ME)
  - Block-based processing:
    - » Macro-block (16 x16) based design.
    - » Block (8x8) can be used as application.
    - » *Progressive* block scanning.
  - *What is the Motion Estimation (ME)*?
    - » Procedure to find the best matching area in the previous picture for the current block (or Macro-block).
    - » A *search range* is generally given or defined by

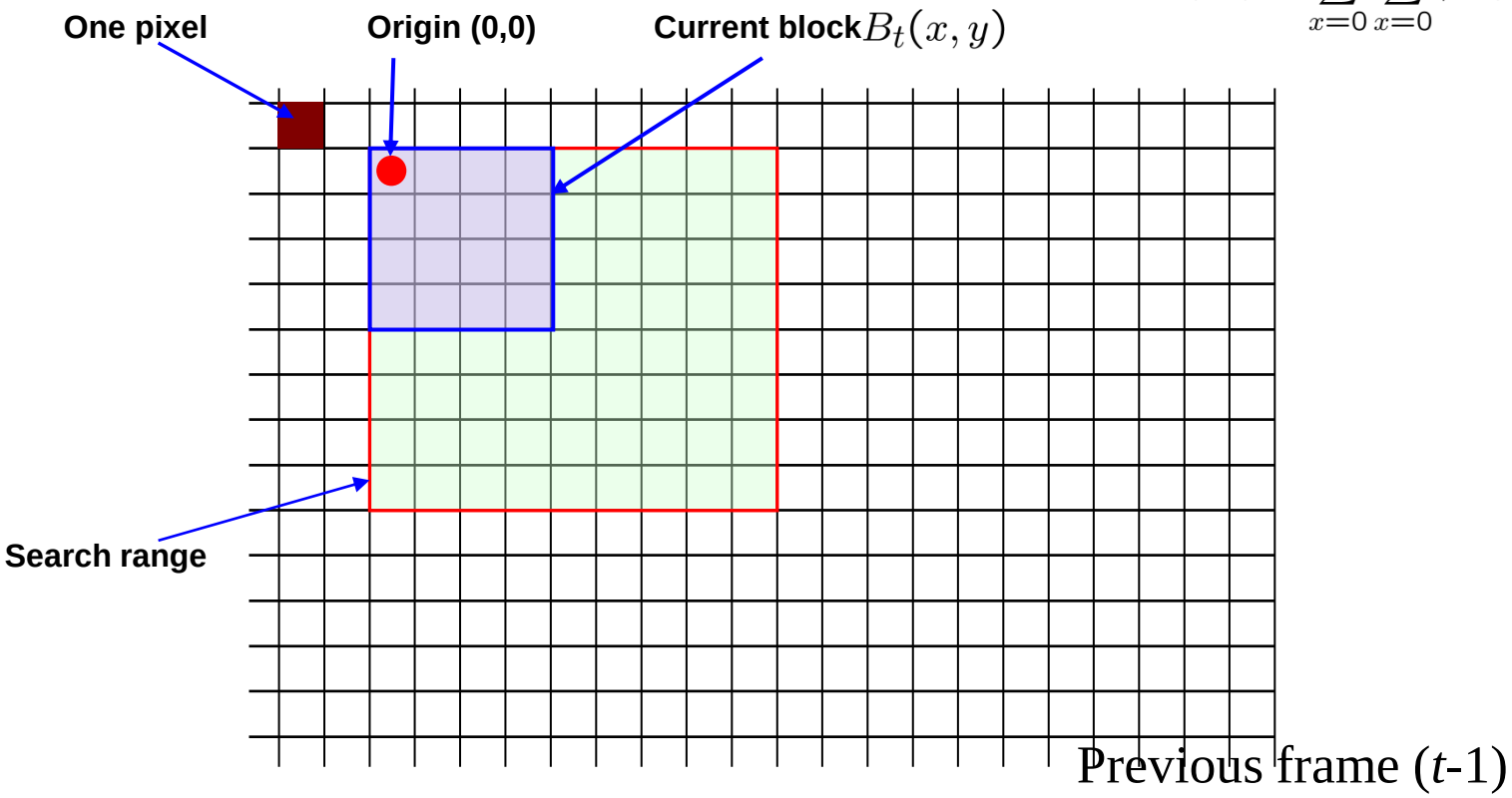
Frame (t-1)

Frame t

Basic Processes for Video Compression

- *The best matching point*:
  - » How to define?
  - » Which process/find?

$$SAD(i,j) = \sum_{x=0}^{x=4} \sum_{x=0}^{x=4} |B_t(x,y) - B_{t-1}(x,y)| \quad (1)$$

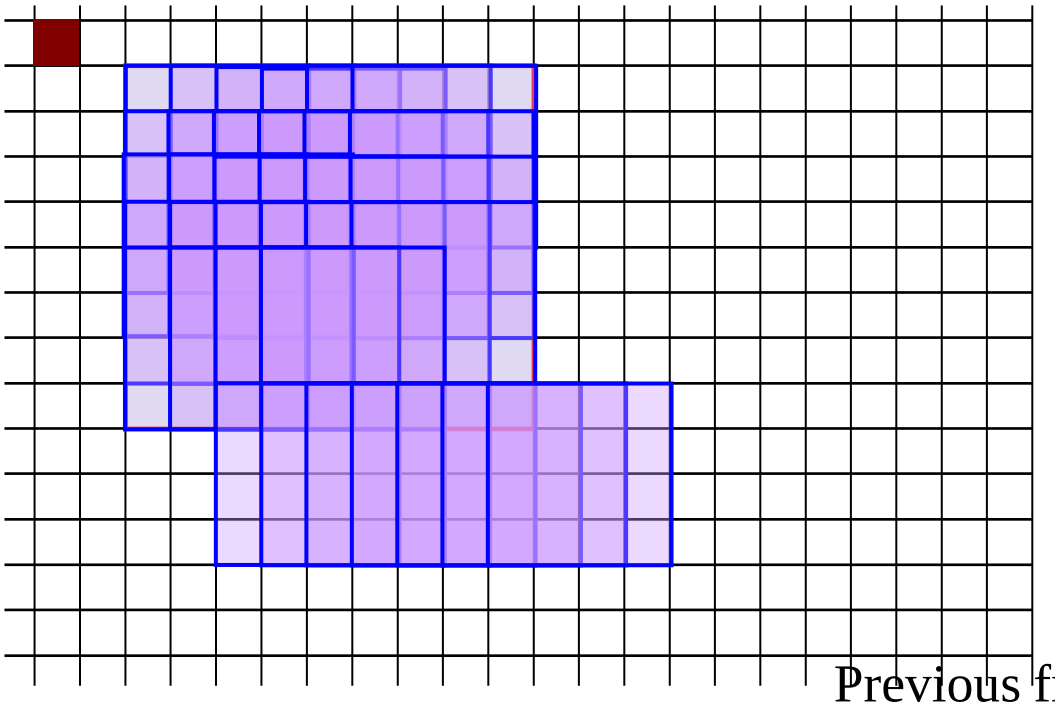


Basic Processes for Video Compression

– *The best matching point*:

» Using *SAD* equation, *SAD* values are computed at every search (pixel) points.

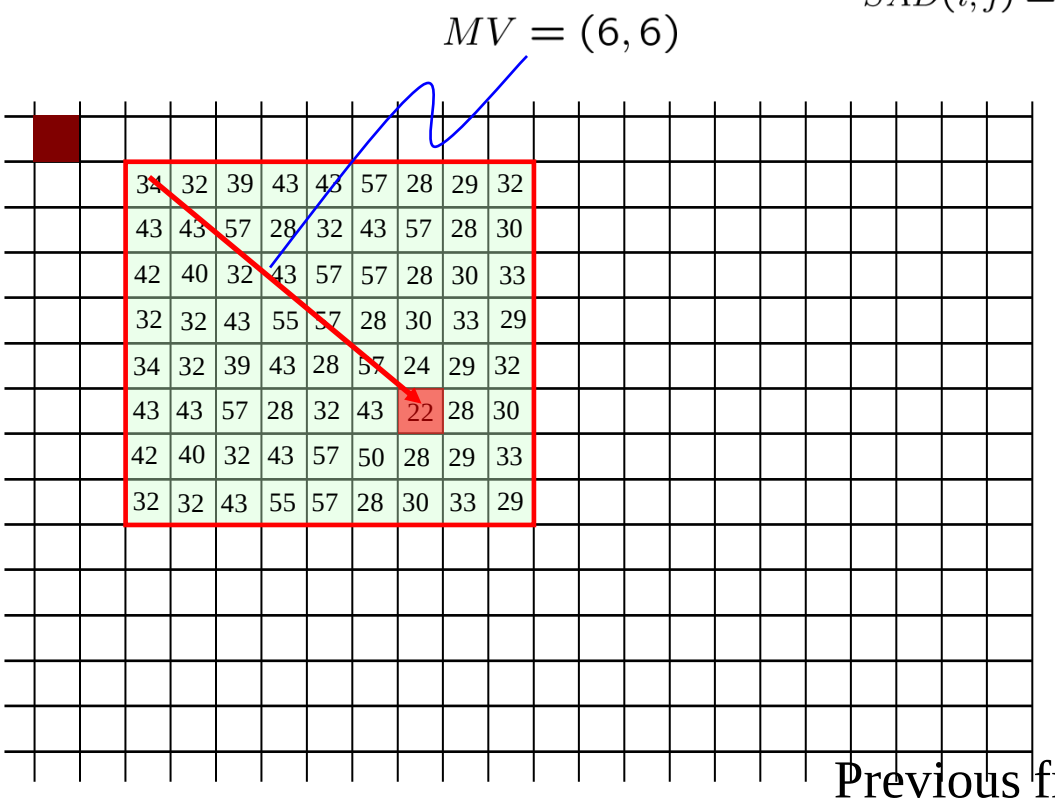
$$SAD(i,j) = \sum_{x=0}^{x=4} \sum_{x=0}^{x=4} |B_t(x,y) - B_{t-1}(x,y)| \quad (1)$$



Previous frame (t-1)

Basic Processes for Video Compression

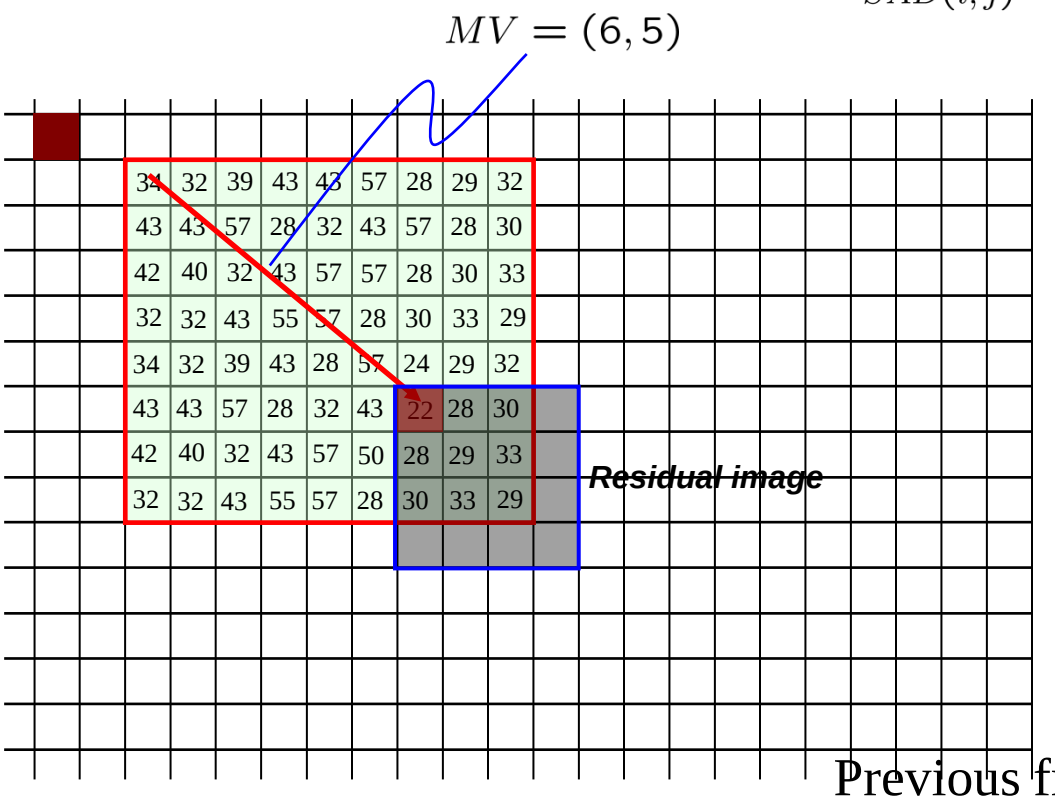
$$SAD(i,j) = \sum_{x=0}^{x=4} \sum_{x=0}^{x=4} |B_t(x,y) - B_{t-1}(x,y)| \quad (1)$$



Previous frame (*t*-1)

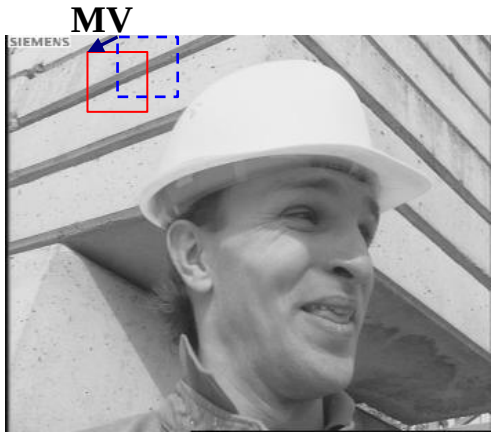
Basic Processes for Video Compression

$$SAD(i,j) = \sum_{x=0}^{x=4} \sum_{x=0}^{x=4} |B_t(x,y) - B_{t-1}(x,y)| \quad (1)$$



Basic Processes for Video Compression

- **Motion Vector** (in Block-based Motion Estimation)
  - *What is a motion vector (MV)*?
    - » Defined as a *vector from the origin of the current block to the origin of the best matching block (area) in the Frame (t-1)* (the previous frame).

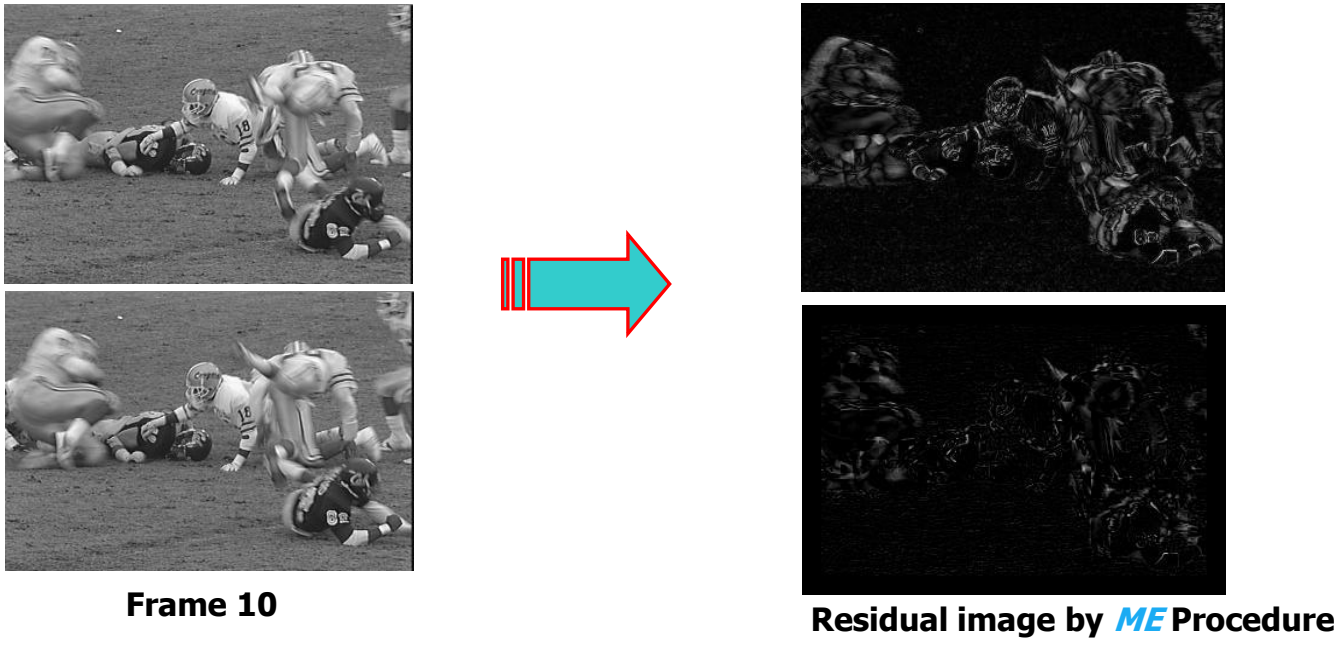


Frame (*t*-1)

Frame *t*

Basic Processes for Video Compression

- Prediction Image (Residual Signal)
  - Subtract the reference area having a minimum distortion from the current block.

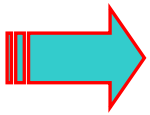


Basic Processes for Video Compression

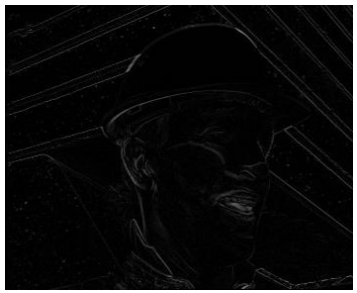
- Prediction Image (Residual Signal)

Frame 9

Frame 10



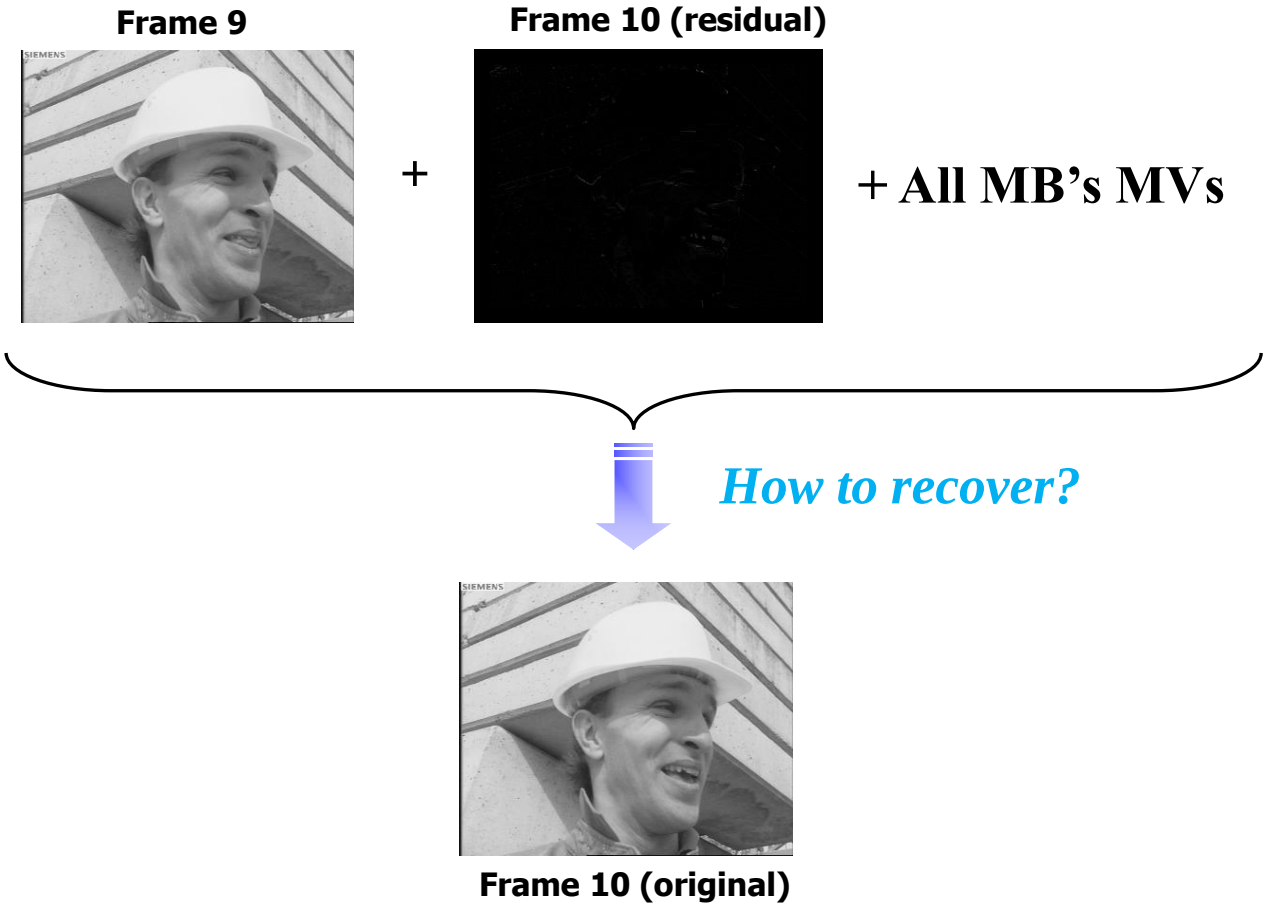
Simple difference image



Residual image by *ME* procedure

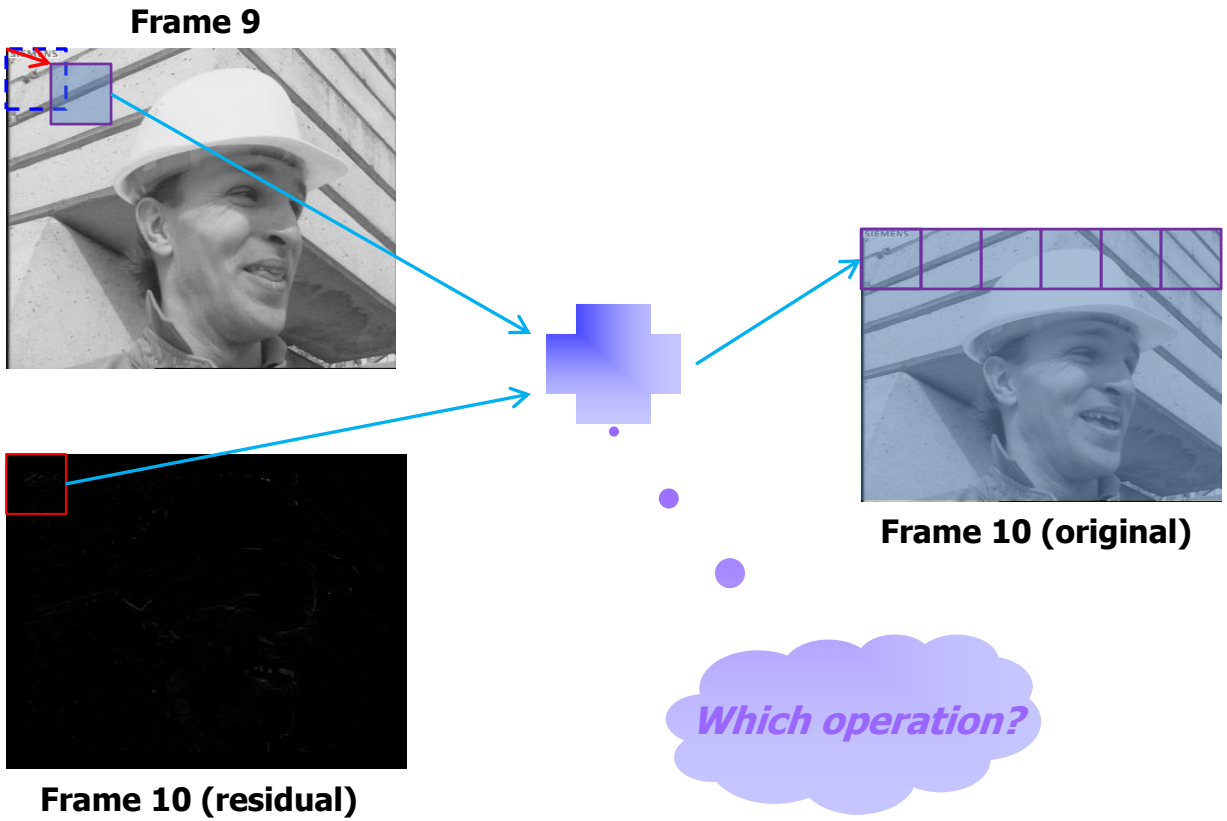
Basic Processes for Video Compression

- Video Compression Format
  - If we have one original frame and its residual frame, we can reconstruct its original frame.



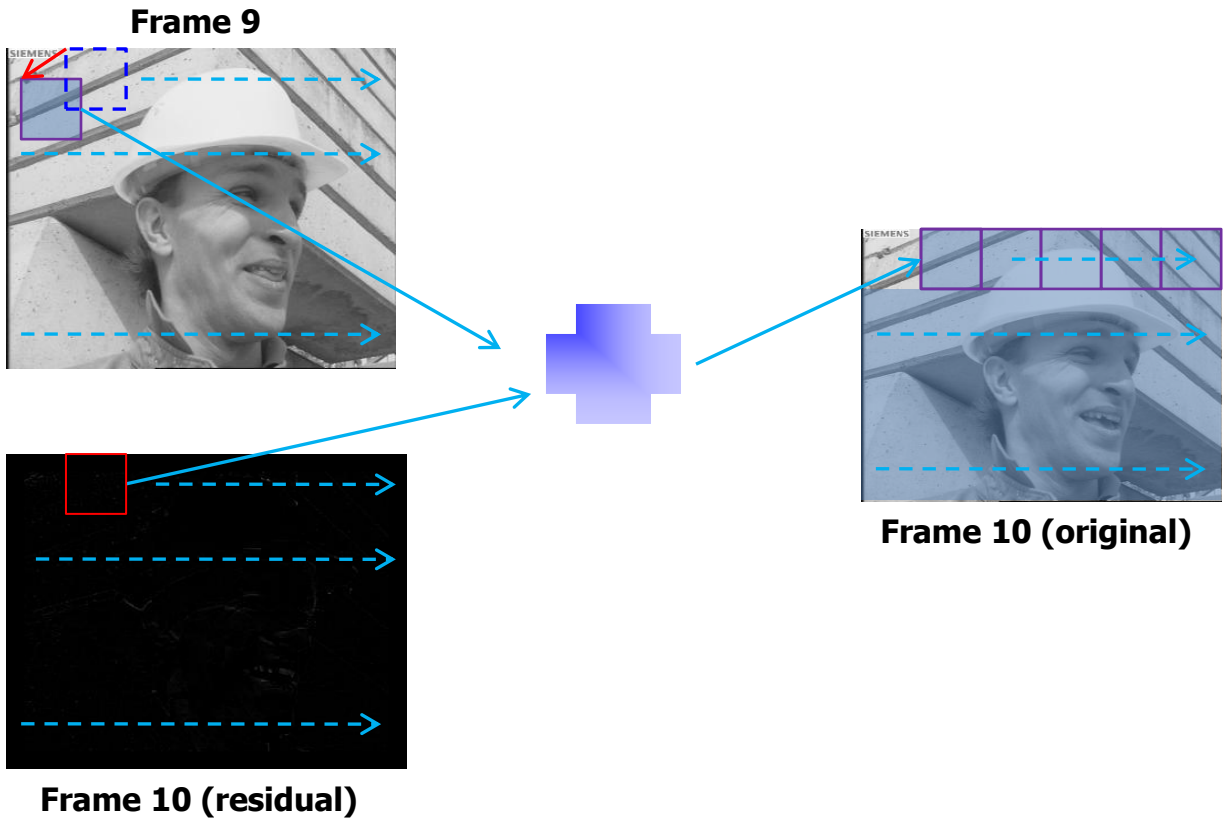
Basic Processes for Video Compression

- Reconstruction (Recovery) process
  - Block (or MB) unit processing.
  - ***Reverse operation of ME procedure, recursively.***



Basic Processes for Video Compression

- Reconstruction (Recovery) process
  - Block (or MB) unit processing.
  - ***Reverse operation of ME procedure, recursively.***

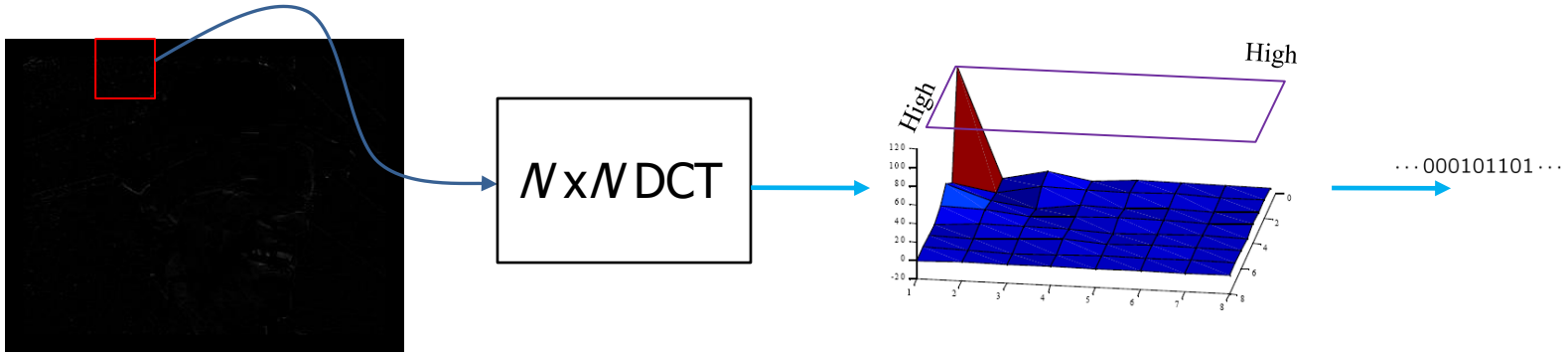


Basic Processes for Video Compression

- DCT for **the residual images** (Block residual images)

By performing ME procedure

  - DCT transformation on the residual images.
    - » Block unit processing.
    - » Energy compaction-based processing.



Basic Processes for Video Compression

- Symbol Coding: Huffman Code (Variable Length Code)
  - Based on frequency of the symbols (coefficients).
    - High frequency symbols: smaller bits ASAP.
    - Low frequency symbols : larger bits ASAP.
    - Variable (bit) length symbol coding.

For example, refer to Table 1-3 on the page 20 in "H.264/AVC TEXTBOOK."

Data Compression Algorithms (1)

- **Huffman coding**
- Run Length Encoding
- Lempel-Ziv-Welch Encoding
- Arithmatic coding
- Delta Encoding

- Some others…
  - Adaptive Huffman coding
  - Wavelet compression
  - **Discrete Cosine Transform**

Data Compression Algorithms (2)

- Huffman Coding
  - The characters in a data file are converted to a binary code.
  - The most common characters in the input file(characters with higher probability) are assigned short binary codes and
  - least common characters(with lower probabilities) are assigned longer binary codes.
  - Codes can be of different lengths

❖ Huffman Coding (Example)

Original source	
Symbol	Probability
a1	0.4
a2	0.3
a3	0.1
a4	0.1
a5	0.06
a6	0.04

- Create a series of source reductions by ordering the probabilities of the symbols under consideration and combining the lowest probability symbols into single symbol that replace them in the next source reduction. This process is then repeated until a reduced source with two symbols is reached.

Data Compression Algorithms (4)

Original source		Source Reduction			
Symbol	Probability	1	2	3	4
a1	0.4	0.4	0.4	0.4	0.6
a2	0.3	0.3	0.3	0.3	0.4
a3	0.1	0.1	0.2	0.3	
a4	0.1	0.1	0.1		
a5	0.06	0.1			
a6	0.04				

- To code each reduced source, starting with smallest source and working back to the original source.

Data Compression Algorithms (5)

Original source		Source Reduction			
Symbol	Probability	1	2	3	4
a1	0.4 1	0.4 1	0.4 1	0.4 1	0.6 0
a2	0.3 00	0.3 00	0.3 00	0.3 00	0.4 1
a3	0.1 011	0.1 011	0.2 010	0.3 01	
a4	0.1 0100	0.1 0100	0.1 011		
a5	0.06 01010	0.1 0101			
a6	0.04 01011				

$$L_{avg} = (0.4)(1)+(0.3)(2)+(0.1)(3)+(0.1)(4)+(0.04)(5)+(0.04)(5)$$
$$= 2.2 \text{ bits/pixel}$$

Huffman's procedure creates the optimal code for a set of symbols and probabilities *subject to the constraint* that the symbols be coded one at a time.

Data Compression Algorithms (6)

- Lempel-Ziv-Welch
  - Uses a dictionary or code table.
  - Done by constructing a "dictionary" of words or parts of words in a message, and then using pointers to the words in the dictionary.
  - LZW to compress text, executable code, and similar data files to about one-half their original size. Higher compressions of 1:5 can also be achievable.

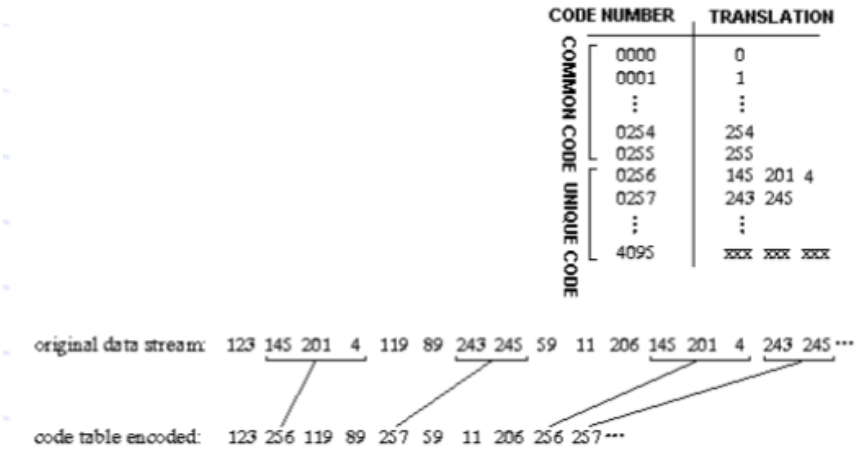
Example:

The rain in the Spain falls mainly on the plain.

- The string "ain" can be stored in the dictionary and then pointed to when it repeats.

Data Compression Algorithms (7)

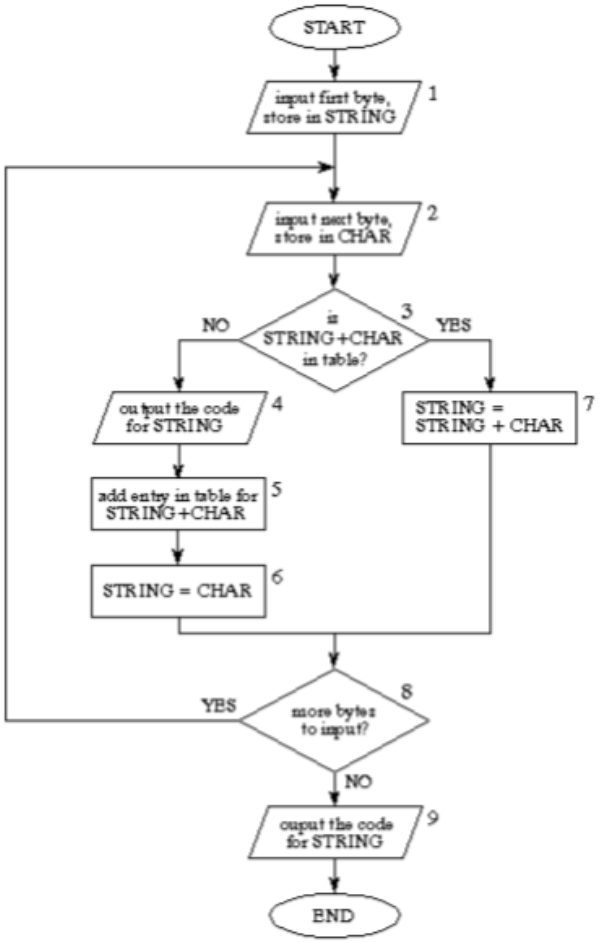
- Code Table Example:



	CHAR	STRING + CHAR	In Table?	Output	Add to Table	New STRING	Comments
1	t	t				t	first character- no action
2	h	th	no	t	256 = th	h	
3	e	he	no	h	257 = he	e	
4	/	e/	no	e	258 = e/	/	
5	EOF	/		/			end of file, output

Data Compression Algorithms (8)

- Procedure:

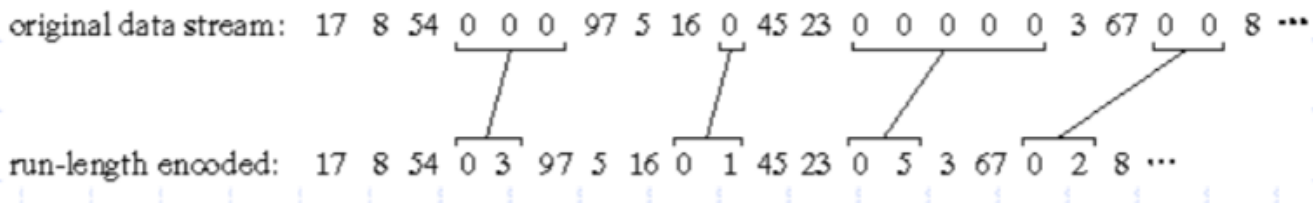


Data Compression Algorithms (9)

- Run Length Encoding (RLE)
  - Coding data with frequently repeated characters.
  - It is called run-length because a run is made for repeated bits and coded in lesser bits by only stating how many bits were there.

  Example:

  - A file with 0 as repeating character.
  - Two characters in the compressed file replace each run of zeros.
  -For the first 3 repeating 0's in original file, the first encdoed stream in compressed file is showing that '0' was repating '3' times.



Data Compression Algorithms (10)

- Arithmetic Coding
  - Message is encoded as a real number in an interval from 0 to 1.
  - Shows better performance than Huffman coding

  - Disadvantages
    - The whole codeword must be received to start decoding.
    - If there is a corrupt bit in the codeword, the entire message could become corrupt.
    - Limited number of symbols to encode within a codeword.

Summary of Basic Processes:

- Spatial Redundancy: DCT (frequency domain)
- Temporal Redundancy: ME (image domain) + DCT(frequency domain)

Hybrid coding

- Symbol Coding (VLC)
- Data Compression Algorithms

- **One question???**
  - **How to process the original (reference) images?**
    - **Based on block processing.**

Frame 9

+

Frame 10 (residual)

+ All MB's MVs

VICL
Visual Information Computing Lab

Thank you for your attention.!!!

QnA

http://vicl.sookmyung.ac.kr