ELEC S347F Multimedia Technologies

Basic Video Representation and Compression

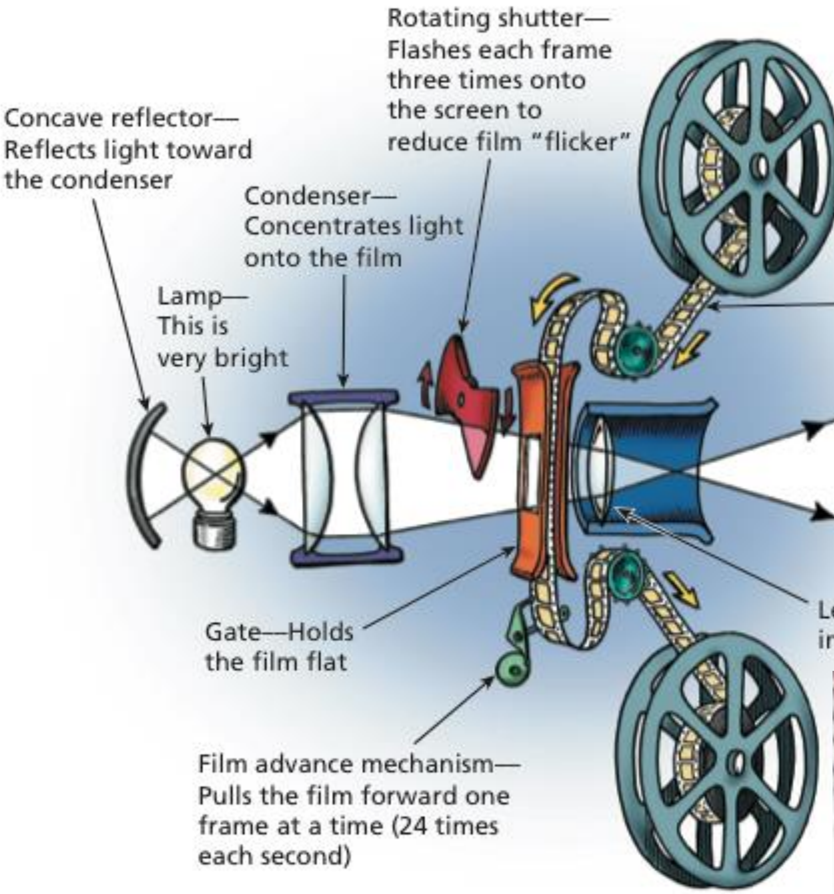
Outline

- Video Basics and Representation
  - Persistence of Vision
  - Phi Phenomenon
  - Frame
  - Interlaced Video
  - Progressive Scan
- Video Format and Compression
  - NTSC & PAL
  - M-JPEG
  - H.261
  - H.263

Persistence of Vision, Phi Phenomenon

- Persistence of Vision
  - When light strikes the retina, the brain retains the impression of that light for about 1/10 to 1/15 second after the source of that light is removed from sight
  - The eye cannot distinguish changes in light that occur faster than this retention period
  - The changes appear to be continuous to human
- Phi Phenomenon
  - When a series of still images are viewed in rapid succession, they are perceived as motion
- The persistence of vision and phi phenomenon together formed the theory of film

Theory of Film


Concave reflector—Reflects light toward the condenser
Lamp—This is very bright
Condenser—Concentrates light onto the film
Rotating shutter—Flashes each frame three times onto the screen to reduce film "flicker"
Spool—Holds the film
Film—Provides the images
Gate—Holds the film flat
Lens—Focuses the image onto the screen
Film advance mechanism—Pulls the film forward one frame at a time (24 times each second)

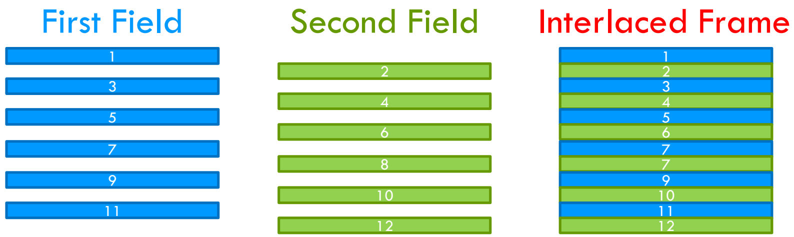
Frame Concept

- A video is composed of sequence of pictures
  - Each picture is called a frame
  - The rate of change of frames is called frame rate
  - The unit is frames per second (fps)
- Whenever the frame rate is higher than the retention rate
  - The sequence of pictures is perceived as motion picture
  - Otherwise it is perceived as discontinuous pictures
  - Movies are typically 24 fps
  - TVs: typically 25 fps (PAL) or 30 fps (NTSC)
  - 4K UHD: typically 60 fps, up to 120 fps

Data Size

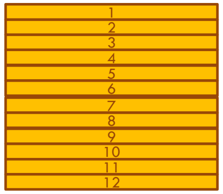
- Raw Video Size = Picture Size x Frame Rate
  - Picture Size = Frame Resolution x Color Depth
  - For example, PAL video resolution is 720 x 576
  - Raw picture size = 720 x 576 x 24 bits ≈ 10 Mb
  - Raw video size = 720 x 576 x 24 bits x 25 fps ≈ 250 Mbps
  - 4K UHDTV: 3840 x 2160@60fps@24bits ≈ 12 Gbps (excluding audio)
- How to reduce the data size?
  - Color Depth
    - Chroma subsampling (similar to JPEG's idea)
  - Frame Resolution
    - Exploit the spatial redundancy in each frame (e.g. JPEG coding)
    - Exploit the temporal redundancy between frames
  - Frame Rate
    - Interlacing, progressive scan

Interlacing



- Instead of transmitting a complete frame, each frame is divided into two fields
  - The even field contains even-numbered horizontal lines, while the old field contains odd-numbered horizontal lines
  - The fields are transmitted alternately: the first pass of a frame transmits the odd field, and the next pass transmits the even field
  - Persistence of vision makes the eye perceive the two fields as a continuous image
- Thus, doubled the perceived frame rate (or halved the bandwidth)
  - e.g. PAL system: 576i50 (or 576i/25)
  - 576 vertical pixels, interlaced frame rate = 25 Hz
  - Interlaced field rate (perceived frame rate) = 50 Hz

Progressive Frame



Progressive Scan

- Transmitting all the lines of a frame in sequence
- Appear smoother than interlacing (flicker does not exists in progressive scanning)
- In principle, require higher the frame rate for progressive scan for the same perception quality as interlacing
- Example: 720p vs. 1080i
  - Progressive scan is denoted by "p"
  - Interlaced is denoted by "i"
  - Bandwidth tradeoff for frame rate or picture size?

Color

- Analog color TV broadcasting uses YUV/YIQ model
  - Digital color TV uses YCbCr
  - Y is Luminance, U, V, I, Q, Cb, Cr are Chrominance
- Why?
  - Compatibility
    - Backward compatible with black/white TV
  - Compression
    - Human visual system are more sensitive to luminance than chrominance
    - Allow to reduce the bandwidth without dropping the perception quality by allocate less bandwidth to chrominance (for analog) or Chroma subsampling (for digital)

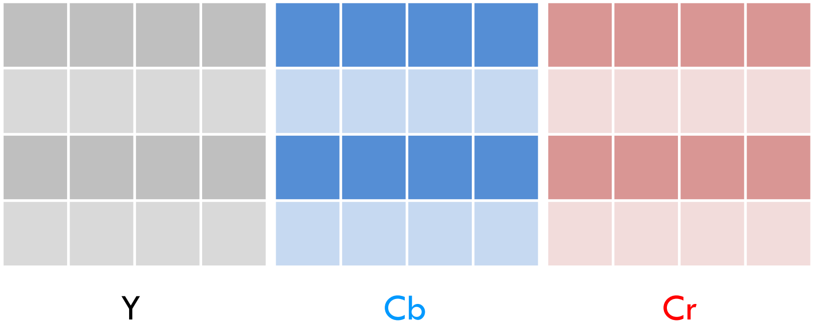
NTSC vs PAL Video Standard

- NTSC
  - Aspect ratio 4:3
  - Interlaced Video
  - 29.97 fps (59.94 fields/s)
  - 525 scan lines per frame
  - $Y + I\cos(\omega t) + Q\sin(\omega t)$
  - 4 MHz is allocated to Y
  - 1.5 MHz to I
  - 0.6 MHz to Q

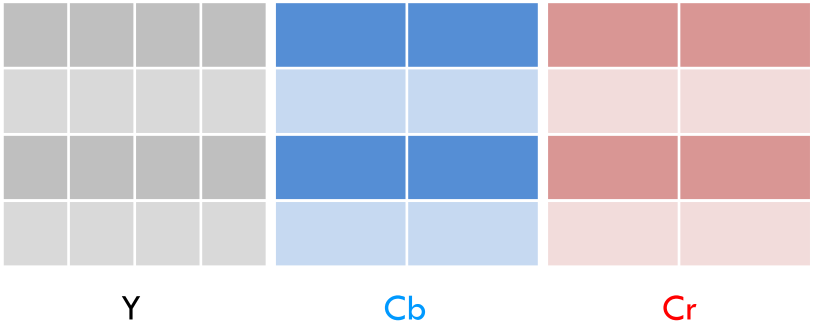
- PAL
  - Aspect ratio 4:3
  - Interlaced Video
  - 25 fps (50 fields/s)
  - 625 scan lines per frame
  - $Y + U\sin(\omega t) + V\cos(\omega t)$
  - 5.5 MHz is allocated to Y
  - 1.8 MHz each to U & V
  - Pros and Cons?

Chroma Subsampling

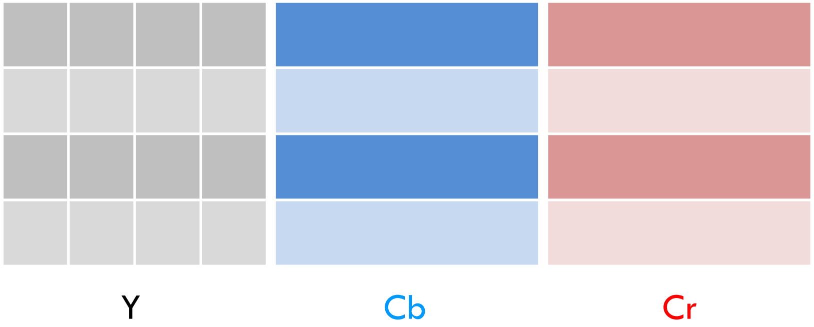
- 4:4:4



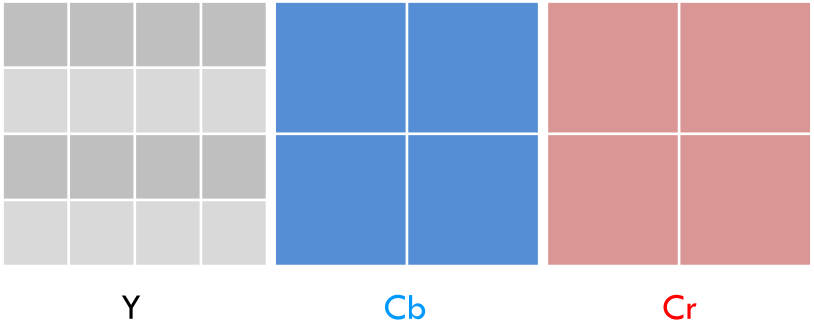
- 4:2:2



- 4:1:1



- 4:2:0



Motion JPEG (M-JPEG)

- Motion JPEG (M-JPEG) is a video format that uses JPEG picture compression for each frame of the video
  - Simple (using existing JPEG algorithm)
  - Used in low-cost IP cameras, digital cameras
- However, M-JPEG is not widely used as a video compression standard
  - Since compressing each frame independently does not exploit temporal redundancy in videos, yielding a poor compression ratio (about 1:20 whereas MPEG-4 1:50)

H.261

Standardization Bodies

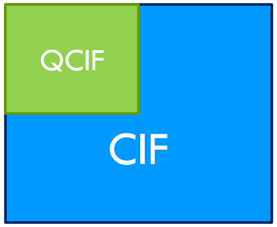
- Since 1985, audio, image and video compression standards have been specified and released by two main groups
  - International Standards Organization (ISO)
    - JPEG, MPEG-1
  - International Telecommunication Union (ITU)
    - H.261, H.263
  - There are some joint development by the groups
    - MPEG-2 Part 2 (H.262), MPEG-4 Part 10 (H.264)

ISO and ITU Standards

- 1989: JPEG issued by ISO (but adopted by ITU as ITU T.81)
- 1991: MPEG-1 released by ISO
- 1993: H.261 released by ITU (based on CCITT 1990 draft)
  - CCITT stands for Comite Consultatif International Telephonique et Telegraphique whose parent organization is ITU
- 1994: H.262 (better known as MPEG-2) released
- 1996: H.263 released
- 1998: MPEG-4 released
- 2002: H.264 released
  - To lower the bit rates with comparable quality video and support wide range of bit rates, and is now part of MPEG 4 (Part 10 Advanced Video Coding (AVC))

H.261 Standard

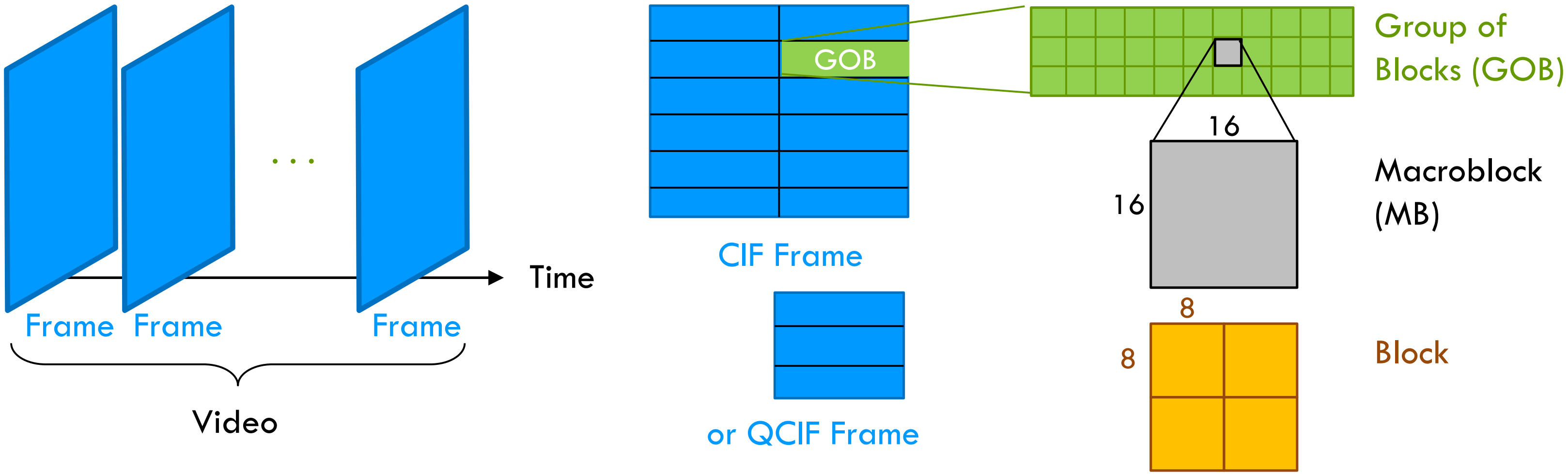
- Designed for video telephony and video conferencing applications
- Intended for use over ISDN telephone lines with constant bit rate
- Support two picture sizes
  - CIF: 352 x 288
  - QCIF: 176 x 144



- YUV color model, 4:2:0 chroma subsampling

H.261 Structure

- A H.261 video is composed of CIF frames
  - Each frame is composed of 2x6 groups of blocks (GOBs) (QCIF frame is composed of 1x3 GOBs)
  - Each GOB is composed of 11x3 macroblocks (MBs)
  - Each MB is composed of 4 8x8 blocks

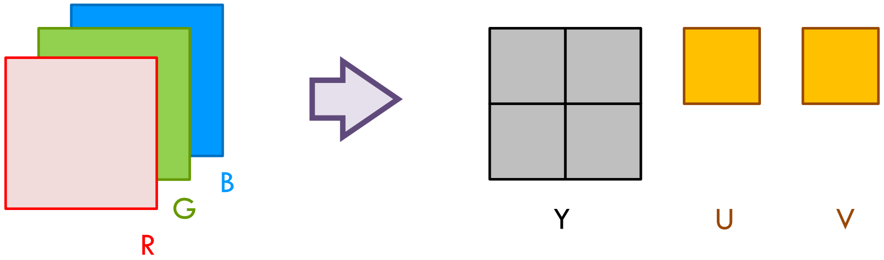


Group of Blocks (GOB)

- Each GOB starts with a sync code
  - 0000 0000 0000 0001
- What is the purpose of GOB?
  - Resynchronization
    - Encoded data are not necessarily start on byte boundaries
    - GOB allows decoder to join in the middle
  - Robustness
    - Avoid error propagation
    - If there is a bit error, decoder can skip and wait until the next GOB to resume decoding

Macroblock (MB)

- Macroblocks and blocks are the basic unit for compression
  - Since H.261 uses 4:2:0 chroma subsampling
  - Each 16x16 pixels of a frame is represented by
    - 1 macroblock + 2 blocks

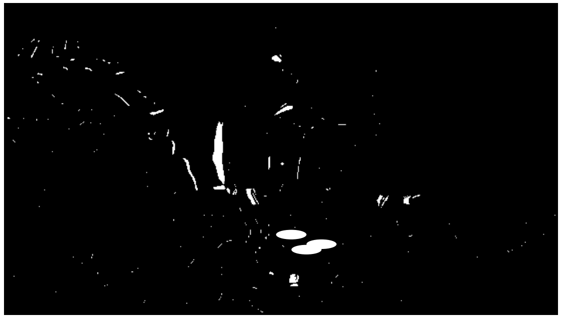


H.261 Intra-Frame Coding

- Each (macro)block is then coded using JPEG algorithm
- Difference between H.261's Intra-frame coding and JPEG coding
  - Larger 16x16 block size for luminance (instead of 8x8 in JPEG)
  - Constant quantization (instead of non-constant matrix)

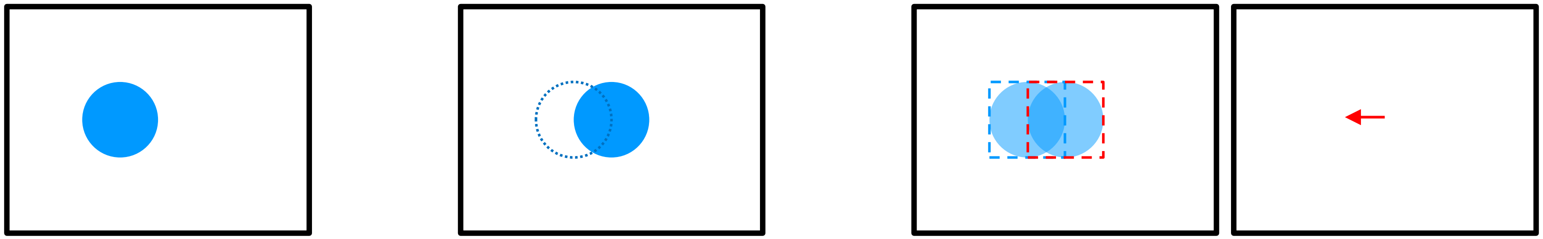
Temporal Redundancy

- How to exploit temporal redundancy in video?
- Most consecutive frames within a sequence are very similar to each other
- The difference between adjacent frames are usually small



Motion Estimation & Compensation

- Encoder: with frame 1 and 2, find a suitable motion vector (i.e. red arrow in below figure) that represents the motion of the moving block



Frame 1 Frame 2 Motion Estimation Motion Vector

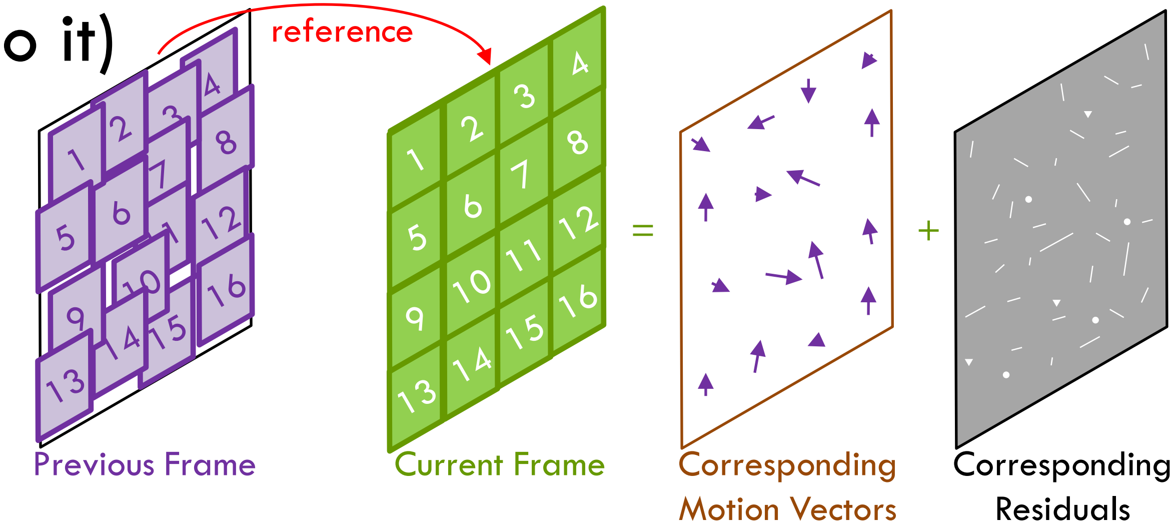
- Decoder: based on frame 1, the motion vector(s) (and corresponding residual, if any), decoder can reconstruct frame 2 by motion compensation

Motion Estimation & Compensation

- Motion Estimation
  - For a certain area (block) of pixels in a picture
  - Find a good estimate of this area in a reference frame within a specified search area
- Motion Compensation
  - Use the motion vectors to compensate the picture
  - Parts of a reference picture can be reused in a subsequent picture
  - Individual parts are spatially compressed (JPEG-like compression)

Block-Based Motion Estimation

- The H.261 standard uses the previous frame to predict the current frame
  - The prediction is done in block sense
  - i.e. for each MB, search the best match one from the previous frame (the standard does not specify how to do it)



Finding the Motion Vectors

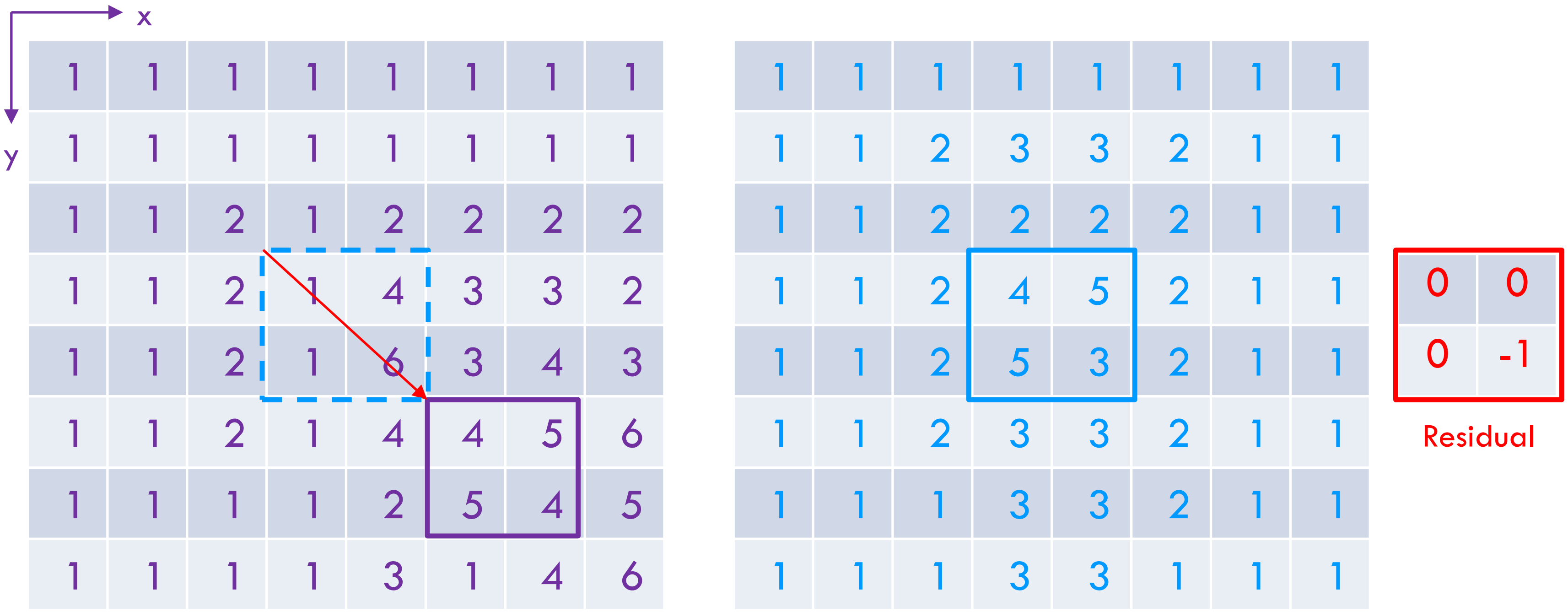
- How to find the motion vectors?
  - For each macroblock (MB), search within a $\pm n \times \pm m$ pixel search window
  - For each search window, compute the sum of absolute differences (SAD) or sum of squared differences (SSD)
  - Finally, the window with minimum SAD/SSD is chosen
  - The vector from the window to the MB is the motion vector

Finding the Motion Vectors

- $\mathrm{SAD}(i, j)$
  - $= \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} |C(x+k, y+l) - R(x+k+i, y+l+j)|$
  - where N is the size of MB
  - C is the original MB, (x, y) is the position of C
  - $C(x+k, y+l)$ is the pixel in the MB with upper left corner (x, y) in the target
  - R is the referenced region
  - $R(x+k+i, y+l+j)$ is the pixel in the MB with upper left corner $(x+i, y+j)$ in the reference
- $\mathrm{SSD}(i, j)$
  - $= \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} (C(x+k, y+l) - R(x+k+i, y+l+j))^2$
- Goal is to find a vector (i, j) such that $\mathrm{SAD}/\ \mathrm{SSD}(i, j)$ is the minimum

Finding the Motion Vectors

- Simplified example: find the motion vector of the below 2x2 MB from the 8x8 picture frame



1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	2	1	2	2	2	2
1	1	2	1	4	3	3	2
1	1	2	1	6	3	4	3
1	1	2	1	4	4	5	6
1	1	1	1	2	5	4	5
1	1	1	1	3	1	4	6

Previous Frame

1	1	1	1	1	1	1	1
1	1	2	3	3	2	1	1
1	1	2	2	2	2	1	1
1	1	2	4	5	2	1	1
1	1	2	5	3	2	1	1
1	1	2	3	3	2	1	1
1	1	1	3	3	2	1	1
1	1	1	3	3	1	1	1

Current Frame

0	0
0	-1

Residual

Min. SAD = |4-4|+|5-5|+|5-5|+|3-4|=1

Motion vector = (2, 2), Residual = {0, 0; 0, -1}

Inter Mode and Intra Mode

- What if min. SAD is too large?
- Define
  - $MB_{mean} = \frac{\sum_{i=0,j=0}^{N-1} |C(i,j)|}{N^2}$
  - $A = \sum_{i=0,j=0}^{N-1} |C(i,j) - MB_{mean}|$
  - If $\text{SAD}_{min} \leq A + 2N^2$, the match is acceptable
    - Inter mode is chosen
  - Otherwise there are no acceptable matches
    - Code that particular MB as an intra MB

Example

- For the previous example, $N = 2$
  - $MB_{mean} = \frac{4+5+5+3}{2^2} = 4.25$
  - $A = |4 - 4.25| + |5 - 4.25| + |5 - 4.25| + |3 - 4.25| = 0.25 + 0.75 + 0.75 + 1.25 = 3$
  - $A + 2N^2 = 3 + 2(2^2) = 11$
  - $\therefore \text{SAD}_{min} = 1 \leq A + 2N^2$
  - i.e. the searched MB is acceptable, so inter mode should be used

Macroblock Coding

- Intra-Frame Compression
  - For MB with no acceptable matches
  - Use JPEG coding except with constant quantization
- Inter-Frame Compression
  - For MB with acceptable match
  - Encode the motion vector (two integers)
  - Encode the residual using the above intra-frame compression

Encoding Motion Vectors

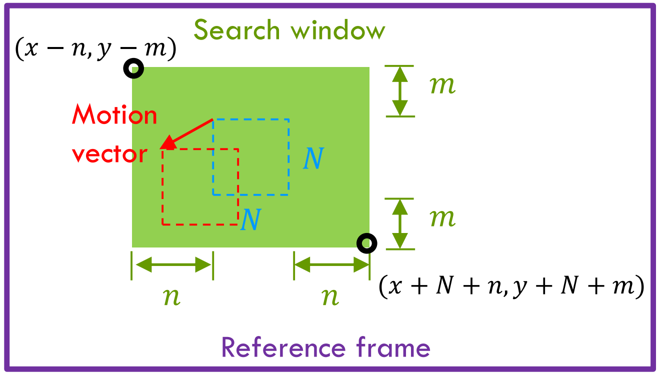
- Motion vectors tend to be highly correlated between MBs
- The horizontal component is compared to the previously valid horizontal motion vector and only the difference is recorded
- Same difference is calculated for the vertical component
- The differential differences are then coded using Huffman code (or other coding) for maximum compression efficiency

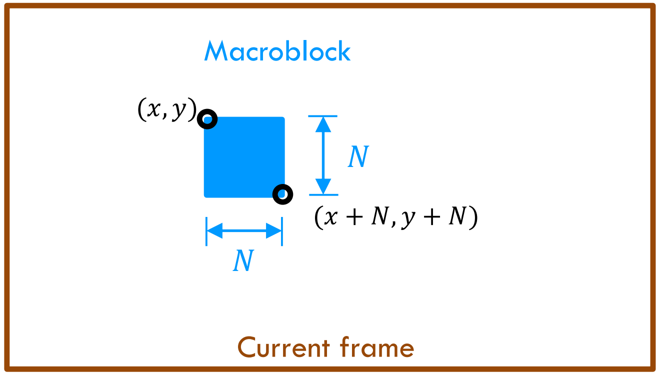
Exhaustive Search for Motion Vector?

- The previous example is a simplified version
- Exhaustive search over a wide 2D area yields the best matching results
  - At extreme computational cost to the encoder
  - The most computationally expensive portion of video encoding
  - It is too costly and is not feasible
  - H.261 uses window search
- Other advanced search methods
  - Logarithmic search
  - Hierarchical search

Window Search

- Search range is limited to $\pm n$ pixels horizontally and $\pm m$ pixels vertically
  - Search step size is s pixel
  - Search the $(2n+1)(2m+1)$ window in the reference frame (need $\frac{(2n+1)(2m+1)}{s^2}$ ops. per MB)
  - For H.261, $n = m = 16, s = 1$



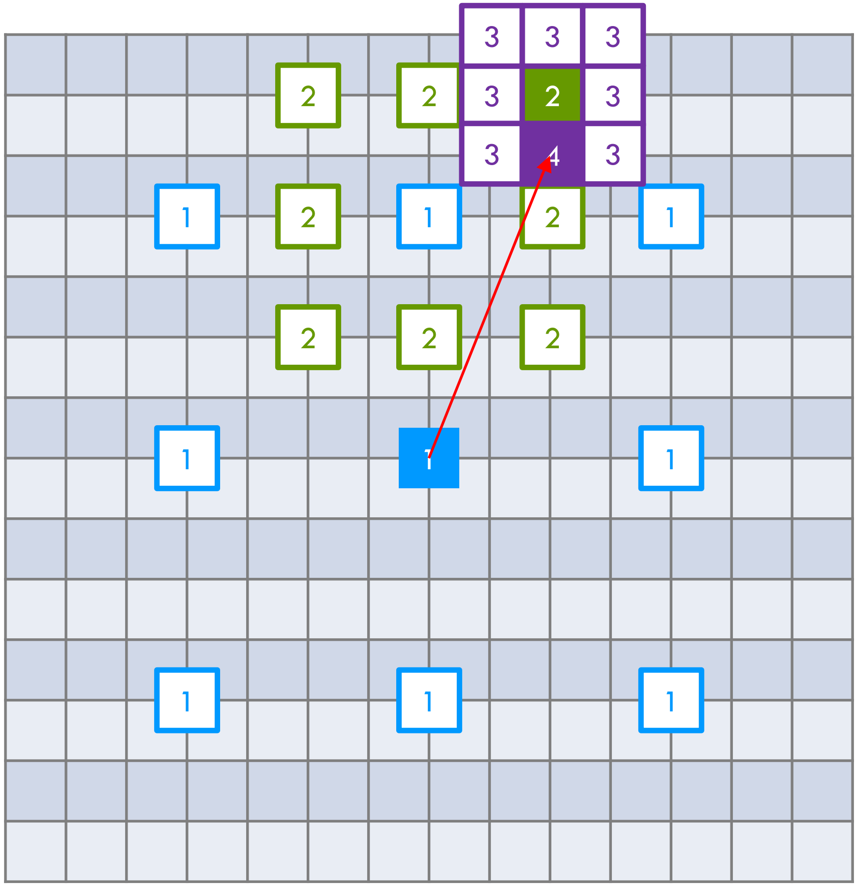


Logarithmic Search

- Algorithm
  1) The search step size s is initialized as $[d/2]$ where d is the search range
  2) Perform a SAD-based search on 9 locations (x, y) and $(x \pm s, y \pm s)$
  3) After finding the (x', y') with the min SAD, update (x, y) as (x', y') and reduce s to $[s/2]$
  4) Repeat step 2 unless $s \leq 1$
- Analysis
  - If the search range is $\pm d$, it requires $[log_2 d]$ iterations to finish and $8[log_2 d] + 1$ SADs are computed
  - A corresponding full search requires $(2d + 1)^2$ SAD checks

Logarithmic Search

Example



The search range is ± 7

The log search iterates for $\lceil log_2 7 \rceil = 3$ times only

8 x 3 + 1 = 25 SADs are computed in order to find the best motion vector

Much faster than full search which requires 15 x 15 = 225 SAD computation

Speedup = 9 in this example

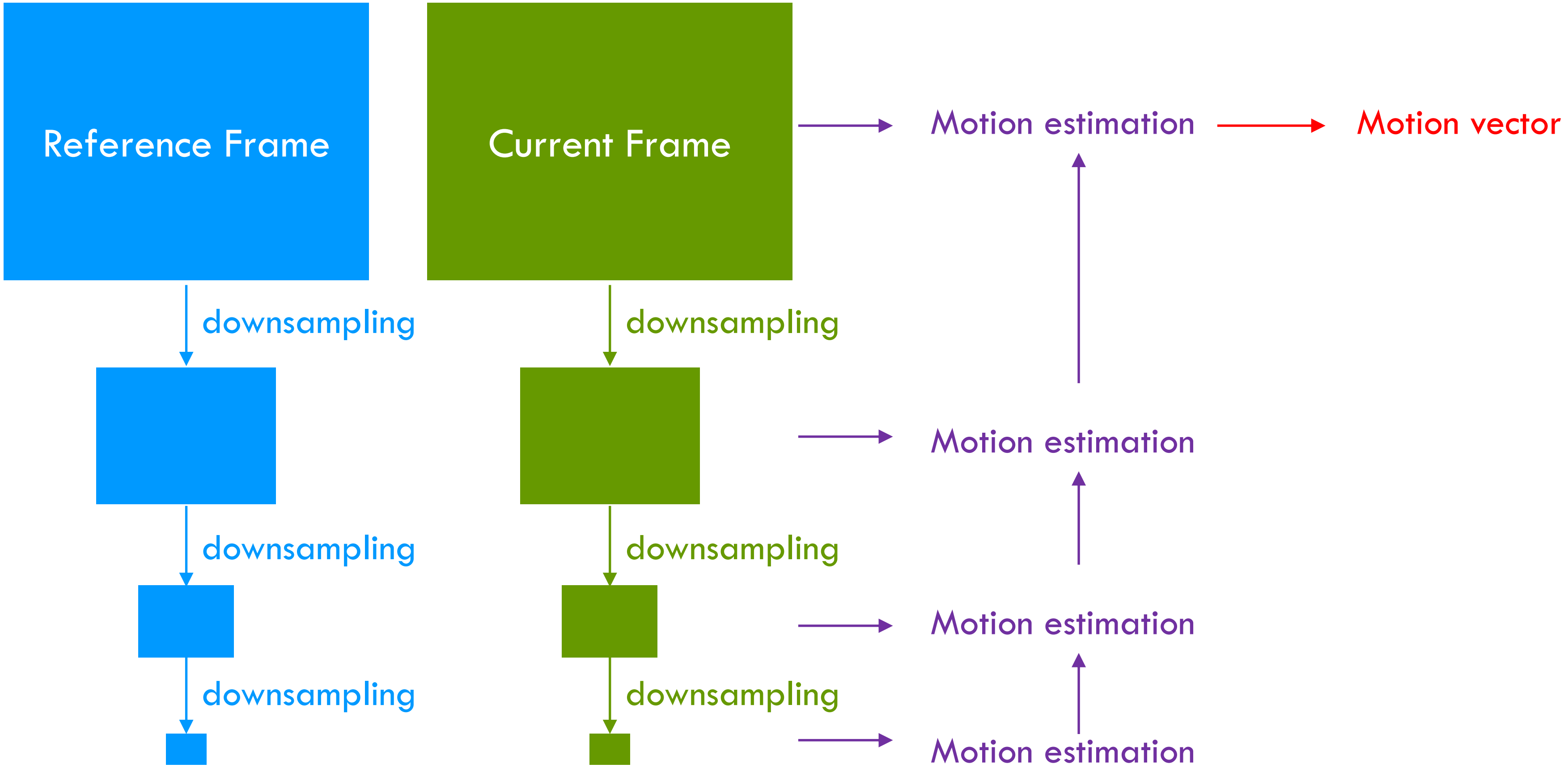
However, the log search is not guarantee to return the best motion vector

Hierarchical Search

- Algorithm
  - Downsample the frame by 2 for k times (where $k > 1$)
  - Find the best match motion vector in the lowest resolution frame
  - Refine the motion vector in a higher resolution frame
  - Repeat until going up to the highest resolution
- Analysis
  - If the picture size is $N \times M$, the lowest resolution frame is of size $\frac{N}{2^k} \times \frac{M}{2^k}$
  - It would be fast even for a full search
  - In total the algorithm requires k window searches to find the best motion vector
  - Higher chance to find the optimal solution than using log search

Hierarchical Search

- Idea



Motion Estimation Summary

- Motion estimation / compensation techniques reduces the video bitrate significantly
- However
  - Introduce extra computational complexity in encoder (time consuming in finding the most appropriate vectors)
  - Encoder / decoder needs extra memory to buffer the previous frames for motion estimation / compensation

Motion Estimation Summary

Search	Computation	Optimal	Remarks
Full	Highest	Yes	Not feasible
Window	Lowest	Suboptimal	Good for low spatial changes frames
Logarithmic	Low	Suboptimal	Not good for frame with multiple similar blocks
Hierarchical	Medium	Suboptimal	Not good for frame with multiple similar blocks Higher chance than window and log search to find the optimal solution High memory requirement for keeping multiple resolution of reference frame and target frame

Limitation of H.261

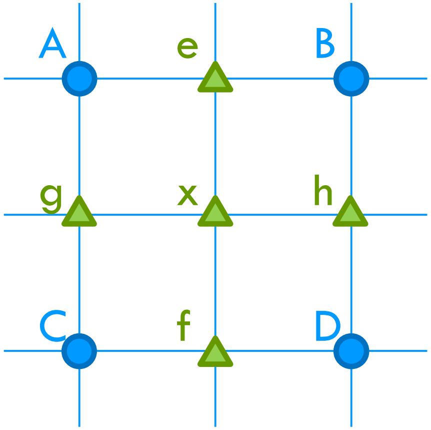
- Limited resolution
  - Support CIF and QCIF only
- Limited robustness
  - The design of GOB protect error being propagate outside GOB
  - However, require the next intra blocks which may appear slowly over next few seconds to recover the damage
- No random access
  - No backward nor forward capability
  - Unless there is a frame with blocks intra-coded entirely
  - Note: H.261 is intended for video conferencing/telephony which backward, forward are not required

H.263

- Improvements in Prediction Accuracy
  - Half-pixel precision in motion vectors
    - i.e. allowing search step size of 0.5 pixel
    - More choices for motion vector
  - Blocks using 8x8 instead of 16x16
    - Small block allows more accurate motion estimation
    - Particular useful for MB contains multiple objects with different motions
- Improvements in Flexibility
  - Support 3 more frame resolution:
  - 16CIF (1408x1152), 4CIF (704x576), SQCIF (128x96)

Bilinear Interpolation

- Original Samples: A, B, C, D
- Half-Pixel Samples: e, f, g, h, x
  - $e = \left\lfloor \frac{A+B}{2} + 0.5 \right\rfloor$
  - $f = \left\lfloor \frac{C+D}{2} + 0.5 \right\rfloor$
  - $g = \left\lfloor \frac{A+C}{2} + 0.5 \right\rfloor$
  - $h = \left\lfloor \frac{B+D}{2} + 0.5 \right\rfloor$
  - $x = \left\lfloor \frac{A+B+C+D}{4} + 0.5 \right\rfloor$



Why adding 0.5?