Video Compression 1: H 261

Multimedia Systems (Module 4 Lesson 2)

Summary:

- H 261 Coding

  Compress color motion video into a low-rate bit stream at following resolutions:

  - QCIF (176 x 144)
  - CIF (352 x 288)
- Inter and Intra Frame coding
- Motion Vectors

Sources:

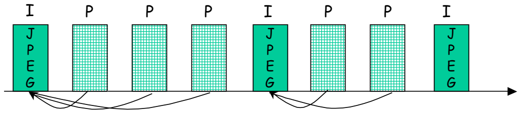
- "Digital Compression for Multimedia: Principles and Standards", Jerry D. Gibson, Toby Berger, Tom Lookabaugh, Dave Lindbergh and Richard L. Baker.
- My research notes

H 261

- The ITU-T recommendation H.261 (a.k.a px64), is for video telephony over ISDN lines.
- H.261 is part of the H.320 group of standards which describes the different components of a video conferencing system and define a narrow-band multimedia terminal.
- H.261 compression algorithm takes advantage of both the spatial and the temporal redundancy of video sequences to achieve high compression ratios.
- H.261 supports low resolution formats due to bandwidth constrains and therefore cannot deliver video broadcast quality; Resolutions:
  - Common Intermediate Format (CIF) 352x288 pixels.
  - Quarter CIF (QCIF) at 176 x 144 pixels.
- The maximum frame rate is 30 frames per second but it can be reduced depending on the application and bandwidth availability

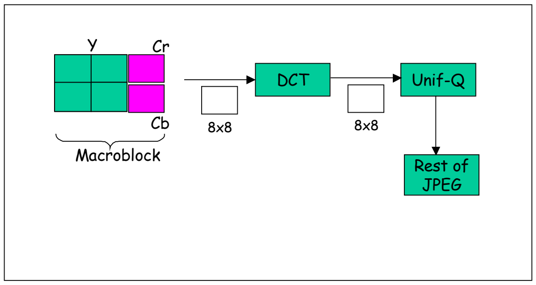
H 261 Coding Basics

- A frame of video is one screenshot.
- Code frames as two types
  - I-frames or Intra-coded frames: Coded by exploiting redundancy within the frame
    - You can think of these as being just the JPEG coding of the frame
    - These are reference points in the video sequence
  - P-frames or Inter-coded frames
    - These are codings of frames that exploit their similarity with previously coded frames
    - Also called *predicted* or *pseudo* frames
- An example H 261 Frame Sequence:



Intra-Frame Coding

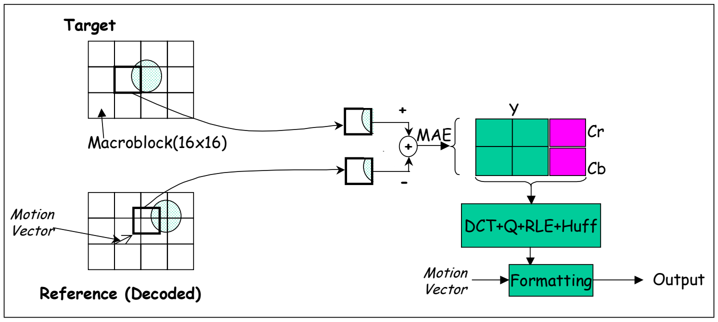
- Macroblocks for coding
  - A Macroblock spans a 16x16 pixel area with 4 Y blocks and one Cr block and 1 Cb block.
    - A block is still 8x8 pixels
  - Uses a uniform Quantization as opposed to a table.



Inter-Frame Coding

Basic Idea:

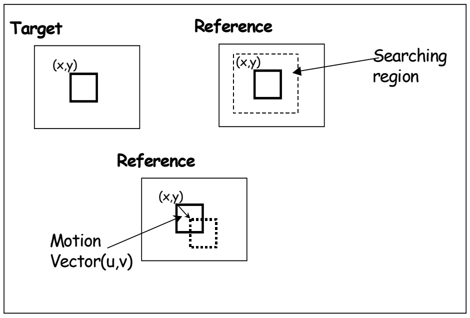
- Encode the "difference" of a motion-compensated part of target frame (frame to code) with respect to a *decoded* reference frame.
  - The reference frame is always the previous I-frame.
- Motion Vector: The "offset" w.r.t to the best match (macroblock) for this macroblock



P-Frame Coding: Motion Vector?

In the previous slide I said, we are looking for a "best" match. How do we do this?

Here is the basic idea:



MAE(Mean Absolute Error):

- C[x..x+15][y..y+15] is the target block
- R[x+u..x+u+15][y+v,y+v+15] is the reference block

$$MAE(u,v) = \frac{1}{N^2} \sum_{i=0}^{15} \sum_{j=0}^{15} |C(x+i, y+j) - R(x+u+i, y+v+j)|$$

Motion Vector?

We search inside the searching region to find a (u,v) such that MAE(u,v) is a minimum.

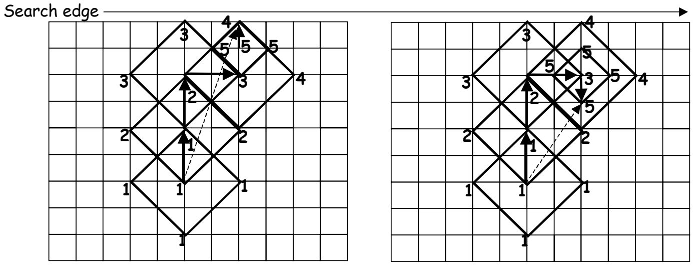
Search Mechanisms

- **Full Search Method**
  - Sequentially search the whole search region (indicated by the dashed area in the figure)
  - Disadvantage: It is an expensive search, hence very slow.
- 2-D Logarithmic search
- Hierarchical Estimation
- Other Enhancements:
  - Relax the integer pixel offset limitation by allowing fractional-pixel offsets instead.

2-D Logarithmic Search

We will discuss two approaches here:

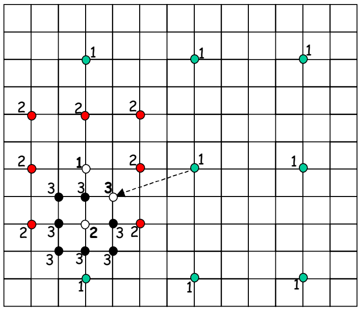
- This approach was published in a paper by Jain and Jain.
  - Iterative comparison of the error measure (MAE) at five neighboring points
  - Logarithmic refinement (divide by 2) of the search pattern **if**
    - Best match is in the center of the 5-point pattern (right figure)
    - OR, center of search pattern touches the border of the search range (left figure).



2-D Logarithmic Search

Second Approach

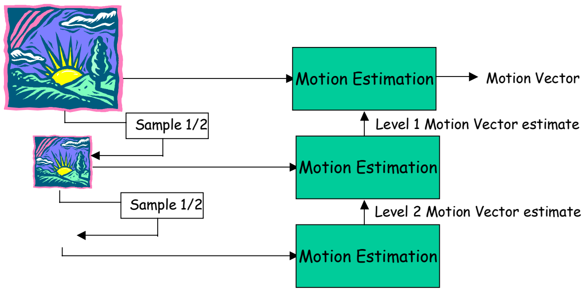
- Iterative comparison of the error measure (MAE) at Nine neighboring points
  - Repeat until the size of the search region is one pixel wide:
    1. Find one of the nine locations that yields the minimum MAE
    2. Form a new searching region with half of the previous size and centered at the location found in step 1.



Hierarchical Estimation

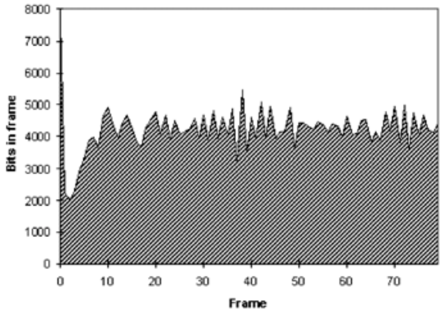
Basic Idea:

- Form a collection of increasingly sub-sampled versions of the current and reference image
- Find a motion vector (*use one of the previous methods) using the lowest resolution images.
- Repeat the same in the higher resolution image using the previous (lower) answer as the initial point for search. Do this until we reach the highest resolution.



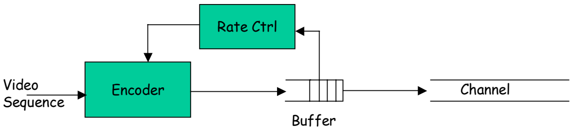
Rate Control

The Problem: H.261 is typically used to send data over a constant bit rate channel, such as ISDN (e.g. 384kbps). The encoder output bit rate varies depending on amount of movement in the scene (see the graph below which shows the bit rate variation for a typical H.261 encoded video sequence). Therefore, a **rate control** mechanism is required to map this varying bit rate onto the constant bit rate channel.

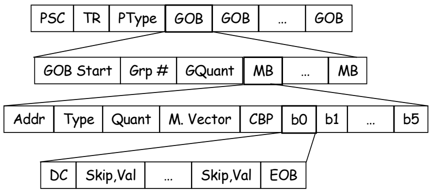


Rate Control: Solution

- The encoded bitstream is **buffered** and the buffer is emptied at the constant bit rate of the channel
- An increase in scene activity will result in the buffer filling up
  - the **quantization step size** in the encoder is increased which increases the compression factor and reduces the output bit rate
- If the buffer starts to empty, then the **quantization step size** is reduced which reduces compression and increases the output bit rate
- The compression, and the quality, can vary considerably depending on the amount of motion in the scene
  - relatively "static" scenes lead to low compression and high quality
  - "active" scenes lead to high compression and lower quality

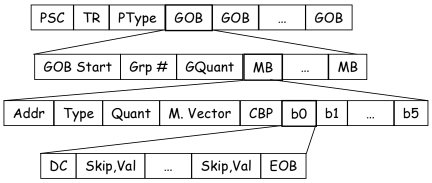


H.261 BitStream Format



- Need to delineate boundaries between pictures, so send Picture Start Code --> *PSC*
- Need timestamp for picture (used later for audio synchronization), so send Temporal Reference --> *TR*
- Is this a P-frame or an I-frame? Send Picture Type --> *PType*
- Picture is divided into regions of 11 x 3 macroblocks called Groups of Blocks --> *GOB*
- Might want to skip whole groups, so send Group Number (*Grp #*)
- Might want to use one quantization value for whole group, so send Group Quantization Value --> *GQuant*
- Overall, bitstream is designed so we can skip data whenever possible while still unambiguous.

H.261 BitStream Format



- Many macroblocks will be exact matches (or close enough). So send address of each block in image --> *Addr*
- Sometimes no good match can be found, so send INTRA block --> *Type*
- Will want to vary the quantization to fine tune compression, so send quantization value --> *Quant*
- Motion vector --> *vector*
- Some blocks in macroblock will match well, others match poorly. So send bitmask indicating which blocks are present (Coded Block Pattern, or *CBP*).
- Send the blocks (4 Y, 1 Cr, 1 Cb) as in JPEG.

H.263

- H. 263 is an improved standard for low bit-rate. Like H. 261, it uses the transform coding for intra-frames and predictive coding for inter-frames.
- Advanced Options:
  - Half-pixel precision in motion compensation
  - Unrestricted motion vectors
  - Syntax-based arithmetic coding
  - Advanced prediction and PB-frames
- In addition to CIF and QCIF, H. 263 could also supports SQCIF, 4CIF, and 16CIF.