CM3106 Multimedia

MPEG Video Compression

Dr Kirill Sidorov
SidorovK@cardiff.ac.uk
www.facebook.com/kirill.sidorov

Prof David Marshall
MarshallAD@cardiff.ac.uk

School of Computer Science and Informatics
Cardiff University, UK

Video compression

We need to compress video (more so than audio/images) in practice since:

1. Uncompressed video (and audio) data are huge.
   In HDTV, the bit rate easily exceeds 1 Gbps — big problems for storage and network communications.
   E.g. HDTV: 1920 x 1080 at 30 frames per second, 8 bits per YCbCr (PAL) channel = 1.5 Gbps.
2. Lossy methods have to be employed since the compression ratio of lossless methods (*e.g.* Huffman, Arithmetic, LZW) is not high enough for image and video compression.

Video compression: MPEG

Not the complete picture studied here!

Much more to MPEG — plenty of other tricks employed.

We only concentrate on some basic principles of video compression:

- Earlier H.261 and MPEG 1 and 2 standards.

with a brief introduction of ideas used in new standards such as H.264 (MPEG-4 Advanced Video Coding).

Image, video, and audio compression standards have been specified and released by two main groups since 1985:

ISO	International Standards Organisation: JPEG, MPEG.
ITU	International Telecommunications Union: H.261–264.

Compression standards

Whilst in many cases one of the groups have specified separate standards there is some crossover between the groups. *E.g.*:

- JPEG issued by ISO in 1989 (but adopted by ITU as ITU T.81)
- MPEG 1 released by ISO in 1991,
- H.261 released by ITU in 1993 (based on CCITT 1990 draft).
  CCITT stands for Comité Consultatif International Téléphonique et Télégraphique whose parent organisation is ITU.
- H.262 (better known as MPEG 2) released in 1994.
- H.263 released in 1996 extended as H.263+, H.263++.
- MPEG 4 released in 1998.
- H.264 releases in 2002 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, or AVC – Advanced Video Coding).

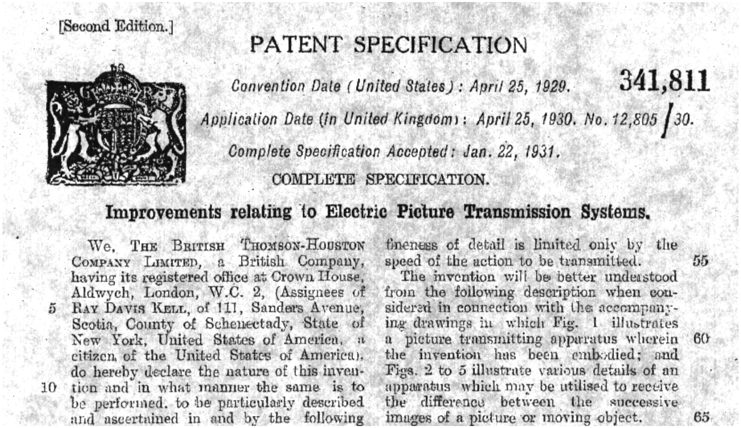
How to compress video?

Basic idea of video compression:
Exploit the fact that adjacent frames are similar.

- Spatial redundancy removal — intraframe coding (JPEG)
  NOT ENOUGH BY ITSELF?
- Temporal redundancy removal — greater compression by using the temporal coherence over time. Essentially we consider the difference between frames.
- Spatial and temporal redundancy removal — intraframe and interframe coding (H.261, MPEG).

Things are much more complex in practice of course.

How to compress video?



[Second Edition.]

PATENT SPECIFICATION

Convention Date (United States): April 25, 1929.

341,811

Application Date (in United Kingdom): April 25, 1930. No. 12,805 / 30.

Complete Specification Accepted: Jan. 22, 1931.

COMPLETE SPECIFICATION.

Improvements relating to Electric Picture Transmission Systems.

We, The British Thomson-Houston Company Limited, a British Company, having its registered office at Crown House, Aldwych, London, W.C. 2, (Assignees of Ray Davis Kell, of 111, Sanders Avenue, Scotia, County of Schenectady, State of New York, United States of America, a citizen of the United States of America), do hereby declare the nature of this invention and in what manner the same is to be performed, to be particularly described and ascertained in and by the following

fineness of detail is limited only by the speed of the action to be transmitted.

The invention will be better understood from the following description when considered in connection with the accompanying drawings in which Fig. 1 illustrates a picture transmitting apparatus wherein the invention has been embodied; and Figs. 2 to 5 illustrate various details of an apparatus which may be utilised to receive the difference between the successive images of a picture or moving object.

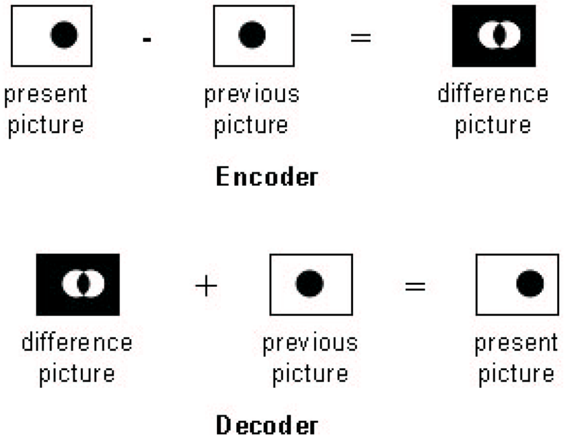
“It has been customary in the past to transmit successive complete images of the transmitted picture.” … “In accordance with this invention, this difficulty is avoided by transmitting only the difference between successive images of the object.”

Simple motion example

Consider a simple image of a moving circle.

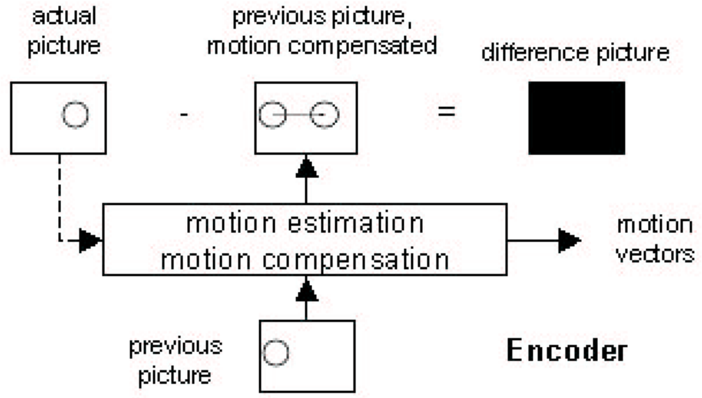
Lets just consider the difference between 2 frames.

It is simple to encode/decode:

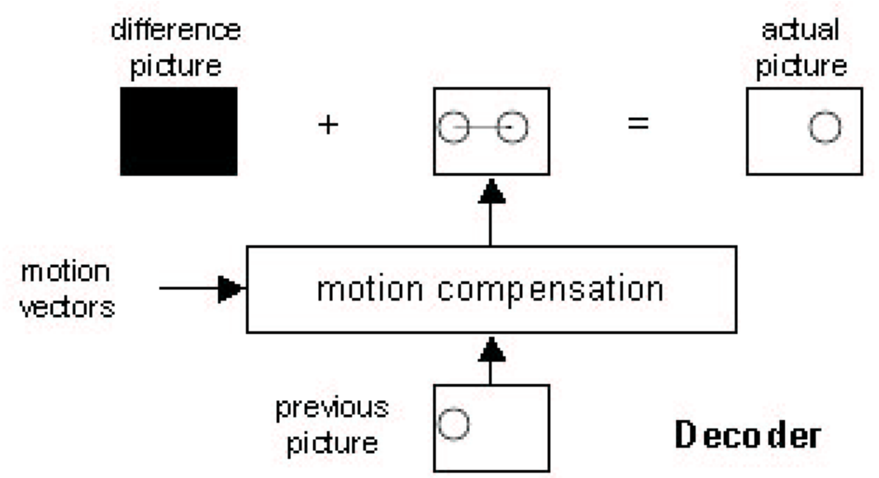


Estimating motion

We will examine methods of estimating motion vectors shortly.

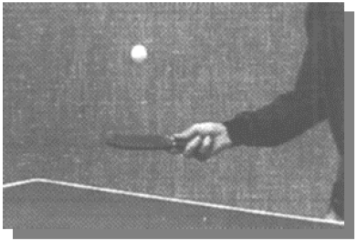


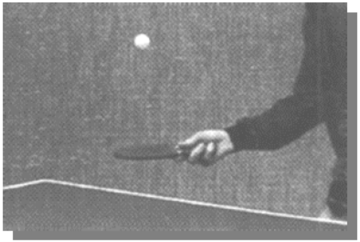
Decoding motion

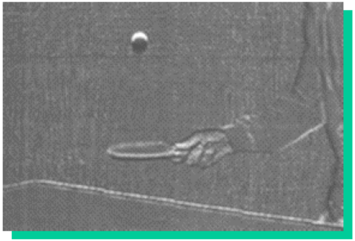


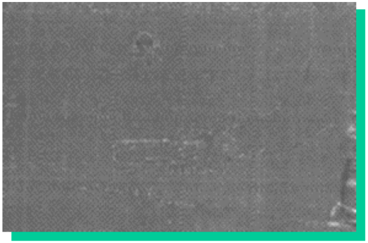
Why is this a better method than just frame differencing?

Motion estimation example


Frame N


Frame N+1


Difference Frame
Without Motion Prediction


Difference Frame
With Motion Prediction

How is motion compensation used?

Block Matching:

- MPEG-1/H.261 relies on block matching techniques.

For a certain area (block) of pixels in a picture:

- Find a good estimate of this area in a previous (or in a future!) frame, within a specified search area.

Motion compensation:

- Uses the motion vectors to compensate the picture.
- Parts of a previous (or future) picture can be reused in a subsequent picture.
- Individual parts spatially compressed — JPEG type compression.

Any overheads?

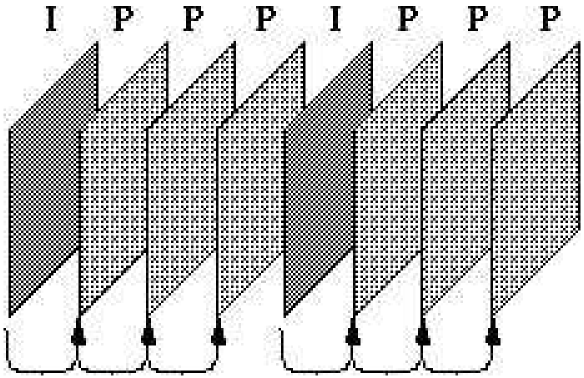
- Motion estimation/compensation techniques reduces the video bitrate significantly
  but
- Introduce extra computational complexity.
  - Decoder needs to buffer reference pictures — backward and forward referencing.
  - Delay.

Lets see how such ideas are used in practice.

Overview of H.261

- Developed by CCITT in 1988-1990 for video telecommunication applications.
- Meant for videoconferencing, videotelephone applications over ISDN telephone lines.
- Baseline ISDN is 64 kbits/sec, and integral multiples ($p \times 64$).
- Frame types are CCIR 601 CIF (Common Intermediate Format) (352×288) and QCIF (176×144) images with 4:2:0 subsampling.
- Two frame types:
  Intraframes (*I-frames*) and Interframes (*P-frames*).
- I-frames use basically JPEG — but YUV (YCrCb) and larger DCT windows, different quantisation.
- I-frames provide us with a refresh accessing point — key frames.
- P-frames use pseudo-differences from previous frame (predicted), so frames depend on each other.

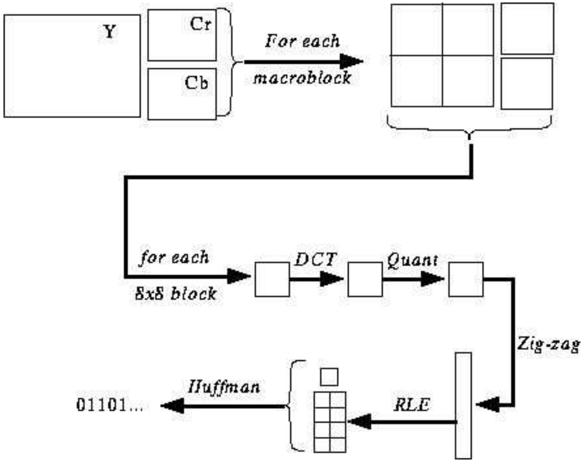
H.261 group of pictures



- We typically have a group of pictures — one I-frame followed by several P-frames — a group of pictures.
- Number of P-frames followed by each I-frame determines the size of GOP — can be fixed or dynamic.
  Why this cannot be too large?

Intra-frame coding

Intra-frame coding is very similar to JPEG:



Intra-frame coding

A basic intra-frame coding scheme is as follows:

- Macroblocks are typically 16x16 pixel areas on Y plane of original image.
- A macroblock usually consists of 4 Y blocks, 1 Cr block, and 1 Cb block. (4:2:0 chroma subsampling)
    - Eye is most sensitive to luminance, less sensitive to chrominance.
    - We operate in a more effective color space: YUV (YCbCr) colour which we studied earlier.
    - Typical to use 4:2:0 macroblocks: one quarter of the chrominance information used.
- Quantization is by constant value for all DCT coefficients.
  I.e., no quantization table as in JPEG.

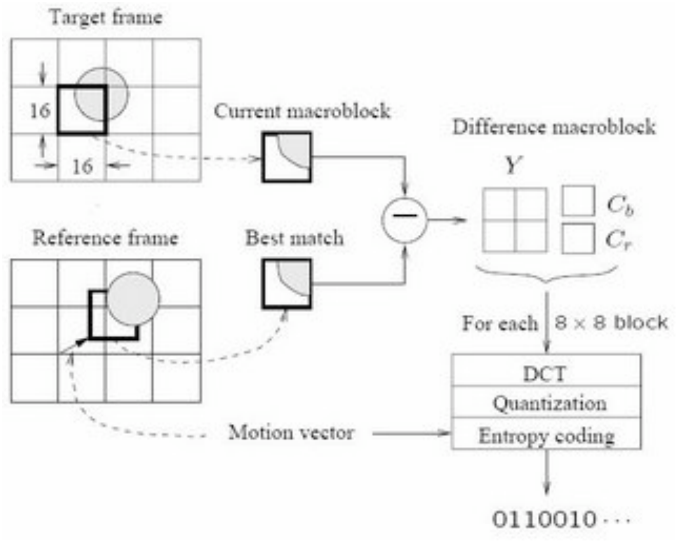
Inter-frame (P-frame) coding

BASIC IDEA:

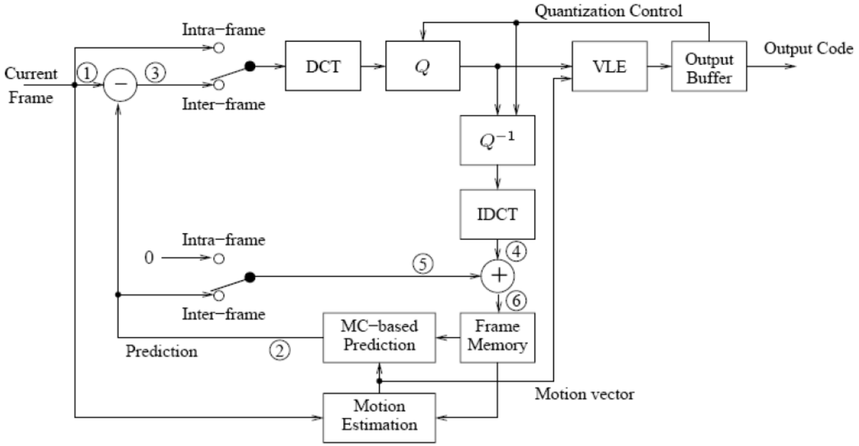
- Most consecutive frames within a sequence are very similar to the frames both before and after the frame of interest.
- Aim to exploit this redundancy.
- Need to use motion estimation.
- Use a technique known as block-based motion compensated prediction.

Inter-frame (P-frame) coding

P-coding can be summarised as follows:

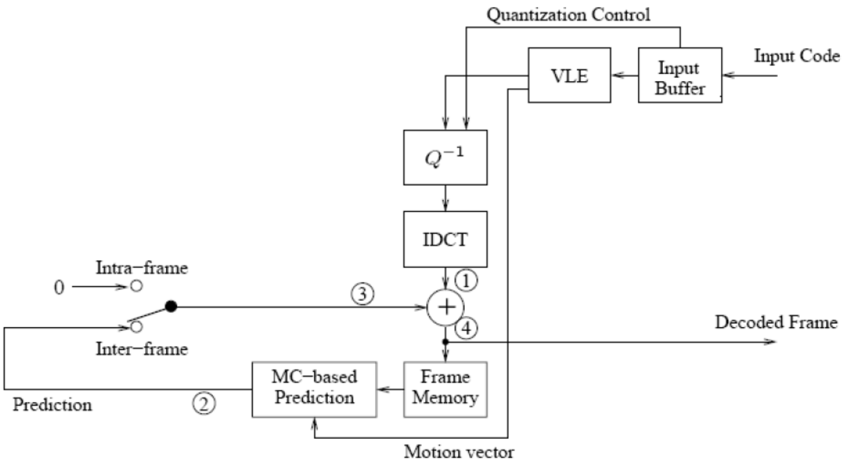


Inter-frame (P-frame) coding



(a) Encoder

Inter-frame (P-frame) coding

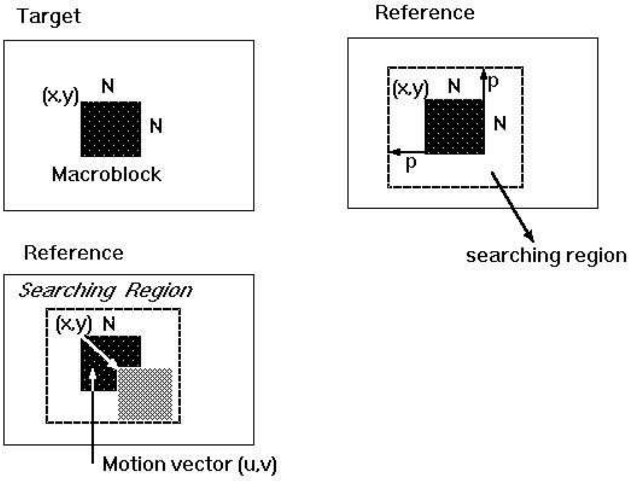


(b) Decoder

Motion vector search

So we know how to encode a P-block.

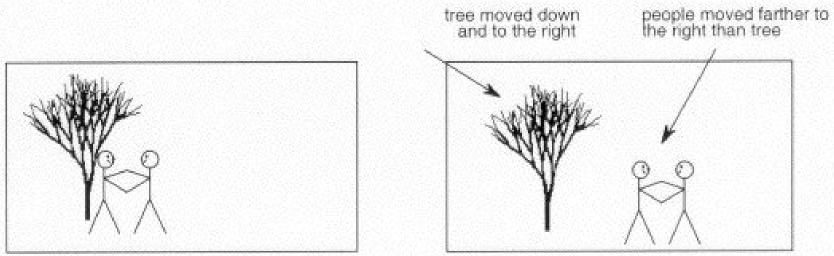
How do we find the motion vector?



Example

The problem for motion estimation to solve is:

- How to adequately represent the changes, or differences, between these two video frames.


tree moved down and to the right
people moved farther to the right than tree

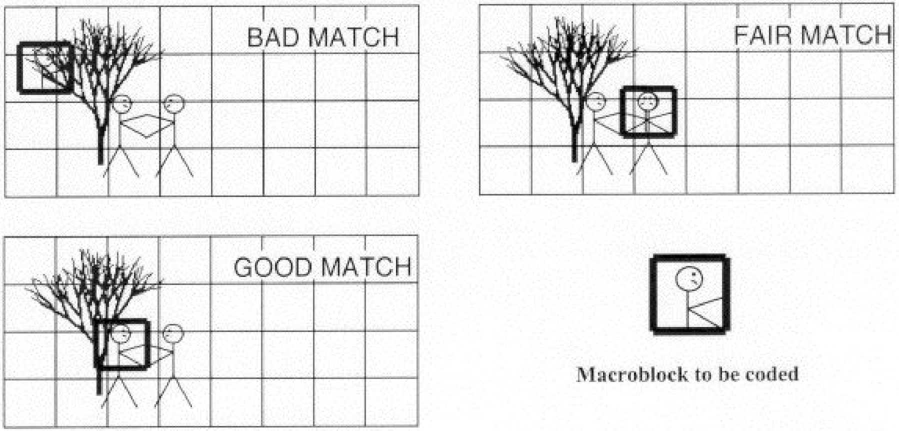
FRAME 1 FRAME 2

Motion estimation

A comprehensive 2-dimensional spatial search is performed for each luminance macroblock.

- MPEG does not define how this search should be performed.
- A detail that the system designer can choose to implement in one of many possible ways.
- Well known that a full, exhaustive search over a wide 2-D area yields the best matching results in most cases, but at extreme computational cost to the encoder.
- Motion estimation usually is the most computationally expensive portion of the video encoding.

Motion estimation example

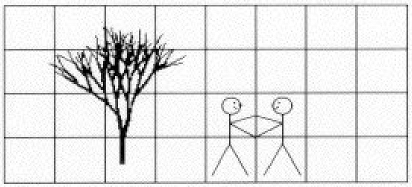


Motion vectors, matching blocks

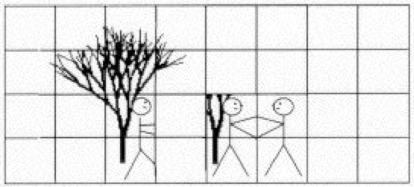
Previous figure shows an example of a particular macroblock from Frame 2 of earlier example, relative to various macroblocks of Frame 1:

- The top frame has a bad match with the macroblock to be coded.
- The middle frame has a fair match, as there is some commonality between the 2 macroblocks.
- The bottom frame has the best match, with only a slight error between the 2 macroblocks.
- Because a relatively good match has been found, the encoder assigns motion vectors to that macroblock,

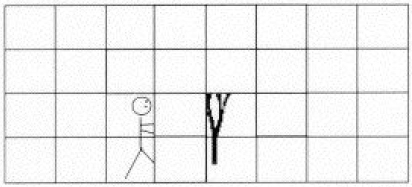
Final motion estimation



Desired Picture

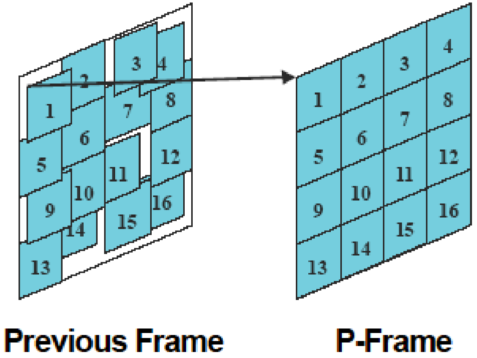


Minus Predicted Picture



Residual Error Picture
(Coded & Transmitted)

Motion estimation



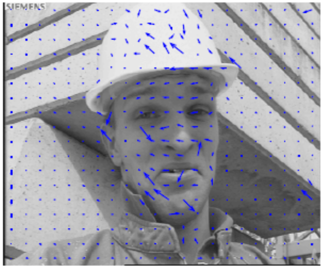
- The predicted frame is subtracted from the desired frame,
- Leaving a (hopefully) less complicated residual error frame which can then be encoded much more efficiently than before motion estimation.

Example

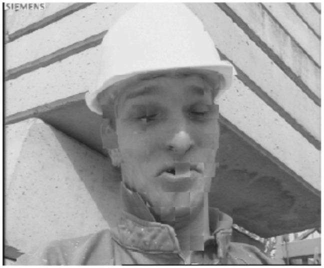
Frame 66

Frame 69

Predicted frame69 with MV overlay

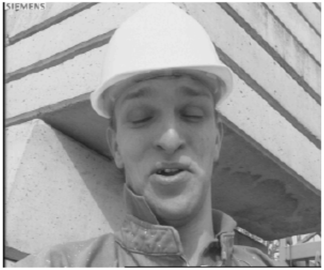


Predicted Frame69

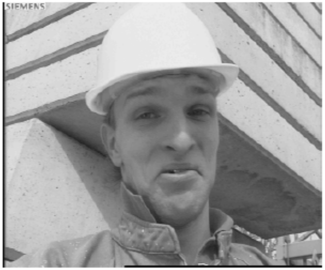


Example

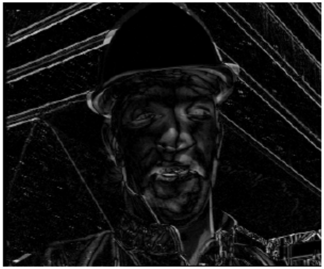
Frame 66



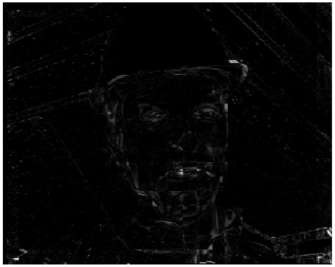
Frame 69



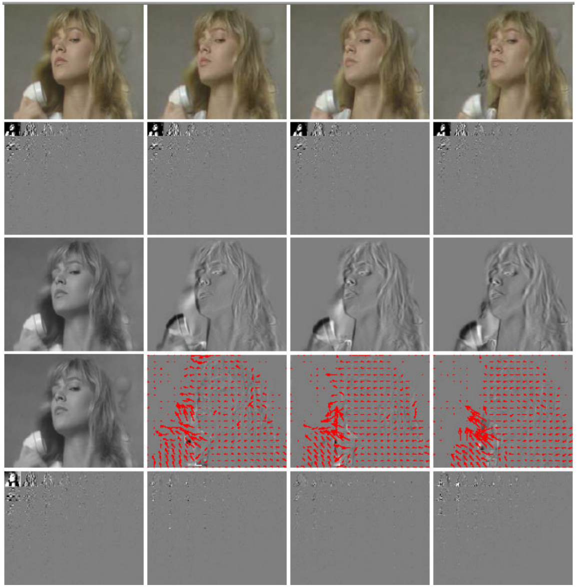
Absolute Difference w/o Motion Compensation



Absolute Difference with Motion Compensation



Example

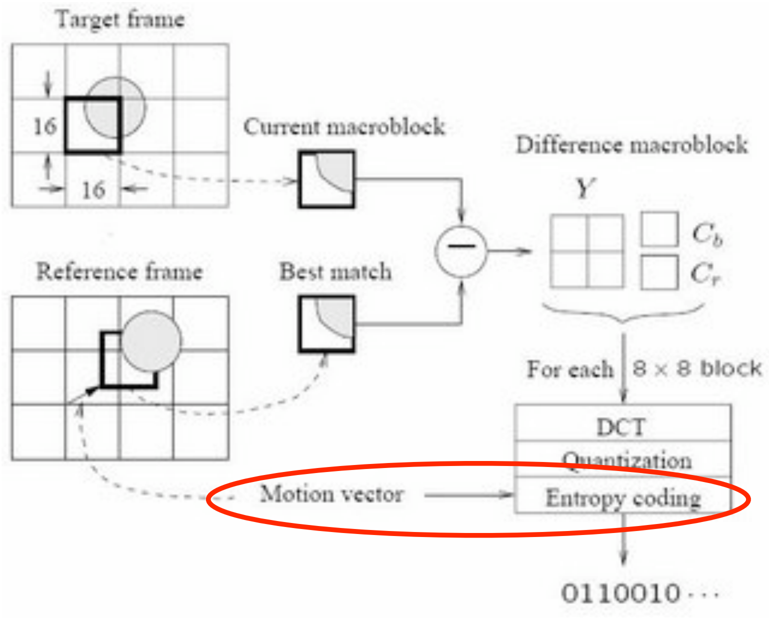


Encoding motion vectors

Differential Coding of Motion Vectors

- Motion vectors tend to be highly correlated between macroblocks:
- The horizontal component is compared to the previously valid horizontal motion vector and
  - Only the difference is coded.
- Same difference is calculated for the vertical component
- Difference codes are then described with a variable length code (*e.g.* Huffman) for maximum compression efficiency.

Recap: P-Frame coding summary



Estimating the motion vectors

So how do we find the motion?

Basic ideas is to search for macroblock.

- Within a $\pm n$ x m pixel search window
- Work out for each window
  Sum of Absolute Difference (SAD)
  (or Mean Absolute Error (MAE))
- Choose window where SAD/MAE is a minimum.

If the encoder decides that no acceptable match exists then it has the option of:

- Coding that particular macroblock as an intra macroblock,
- Even though it may be in a P frame!
- In this manner, high quality video is maintained at a slight cost to coding efficiency.

Sum of absolute differences (SAD)

SAD is computed by:

$$\mathrm{SAD}(i, j) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} |C(x+k, y+l) - R(x+k+i, y+l+j)|$$

- N = size of macroblock window typically (16 or 32 pixels),
- (x, y) the position of the original macroblock, C, and
- R is the reference region to compute the SAD.
- $C(x+k, y+l)$ — pixels in the macro block with upper left corner (x, y) in the target.
- $R(x+k+i, y+l+j)$ — pixels in the macro block with upper left corner $(x+i, y+j)$ in the reference.

Sum of squared differences (SSD)

- Alternatively: sum of squared differences

$$\mathrm{SSD}(i, j) =$$

$$\sum_{k=0}^{N-1} \sum_{l=0}^{N-1} (C(x+k, y+l) - R(x+k+i, y+l+j))^2$$

- Goal is to find a vector (i, j) such that SAD/SSD (i, j) is minimum.

Full search

- Search exhaustively the whole $(2R+1) \times (2R+1)$ window in the reference frame.
- A macroblock centred at each of the positions within the window is compared to the macroblock in the target frame pixel by pixel and their respective SAD (or MAE) is computed.
- The vector (i, j) that offers the least SAD (or MAE) is designated as the motion vector for the macroblock in the target frame.
- Full search is very costly.

Complexity of full search

- Assumptions
  - Block size $N \times N$ and image size $S = M_1 \times M_2$.
  - Search step size is 1 pixel.
  - Search range $\pm R$ pixels both horizontally and vertically.
- Computation complexity
  - Candidate matching blocks $= (2R+1)^2$.
  - Operations for computing MAD for one block $= O(N^2)$.
  - Operations for MV estimation per block $= O((2R+1)^2 N^2)$.
  - Blocks $= S/N^2$.
  - Total operations for entire frame $O((2R+1)^2 S)$.
    - I.e. overall computation load is independent of block size!

Example: M=512, N=16, R=16, 30fps

Approximately 8.55 x 10^9 operations per second!

Real time estimation is difficult. Speed up with GPU?

Full search

Advantages:

- Guaranteed to find optimal motion vector within search range.

Disadvantages:

- Can only search among finitely many candidates. What if the motion is in fractional number of pixels?
- High computation complexity: $O((2R+1)^2S)$.

HOW TO IMPROVE?

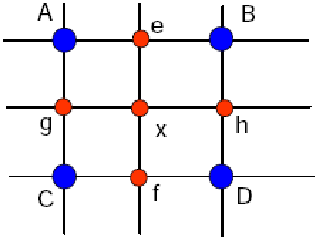
Accuracy: consider fractional translations.

- This requires interpolation (*e.g*. bilinear in H.263).

Speed: try to avoid checking unlikely candidates.

Bilinear interpolation

- Bilinear interpolation:



- Original samples:
  - A, B, C, D
- Half-pixel locations:
  - e, f, g, h, x

$$e = \left\lfloor \frac{A+B}{2} + 0.5 \right\rfloor$$

$$f = \left\lfloor \frac{C+D}{2} + 0.5 \right\rfloor$$

$$g = \left\lfloor \frac{A+C}{2} + 0.5 \right\rfloor$$

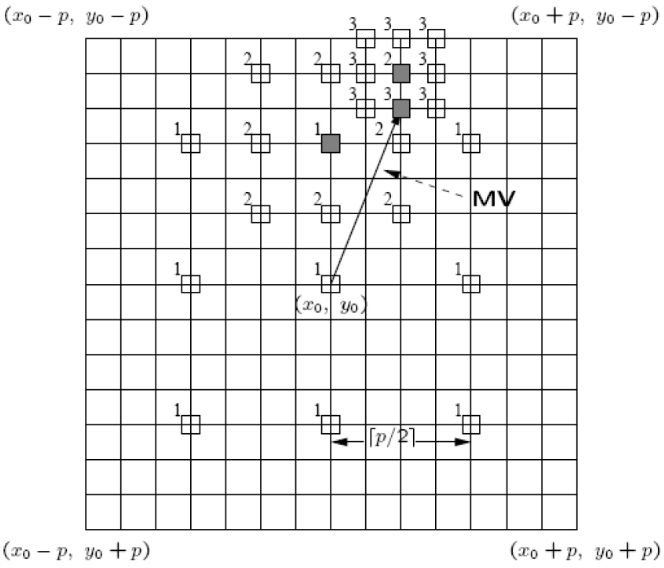
$$h = \left\lfloor \frac{B+D}{2} + 0.5 \right\rfloor$$

$$x = \left\lfloor \frac{A+B+C+D}{4} + 0.5 \right\rfloor$$

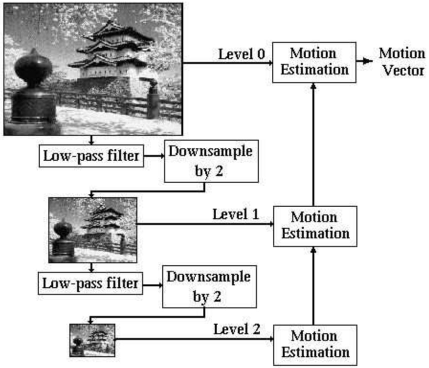
Logarithmic search

- An approach takes several iterations akin to a binary search.
- Computationally cheaper, suboptimal but usually effective.
- Initially only nine locations in the search window are used as seeds for a SAD-based search (marked as ‘1’).
- After locating the one with the minimal SAD, the centre of the new search region is moved to it and the step-size (“offset”) is reduced to half.
- In the next iteration, the nine new locations are marked as ‘2’ and this process repeats.
- If L iterations are applied, for altogether 9^L positions, only $9L$ positions are checked.

Logarithmic search

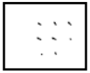


Hierarchical motion estimation

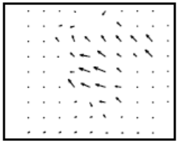


1. Form several low resolution version of the target and reference pictures.
2. Find the best match motion vector in the lowest resolution version.
3. Modify the motion vector level by level when going up.

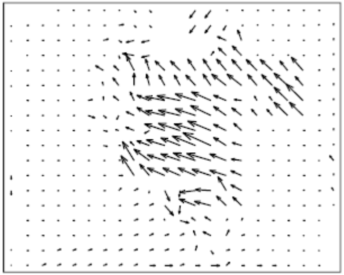
Hierarchical motion estimation

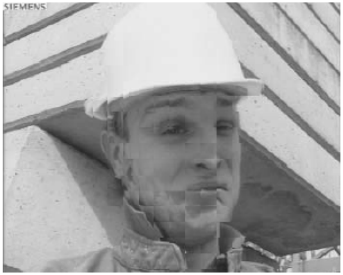

(a)

(b)


(c)

(d)


(e)



(f)

Example: Three-level HBMA

Performance comparison

Operation for 720x480 at 30 fps (GOPS):

Search Method	p = 15	p=7
Full Search	29.890	6.990
Logarithmic	1.020	0.778
Hierarchical	0.507	0.399

Selecting intra/inter frame coding

Based upon the motion estimation a decision is made on whether intra or inter coding is made.

To determine intra/inter mode we do the following calculation:

$$\mathrm{MB}_{\mathrm{mean}} = \frac{\sum_{i=0,j=0}^{N-1} |C(i,j)|}{N^2}$$

$$A = \sum_{i=0,j=0}^{N-1} |C(i,j) - \mathrm{MB}_{\mathrm{mean}}|$$

If $A < (\mathrm{SAD} - 2N^2)$ intra mode is chosen.

MPEG Compression

MPEG stands for:

- Motion Picture Expert Group — established circa 1990 to create standard for delivery of audio and video
- MPEG-1 (1991).Target: VHS quality on a CD-ROM (320 x 240 + CD audio @ 1.5 Mbits/sec).
- MPEG-2 (1994): Target Television Broadcast.
- MPEG-3: HDTV but subsumed into an extension of MPEG-2.
- MPEG 4 (1998): Very Low Bitrate Audio-Visual Coding, later MPEG-4 Part 10 (H.264) for wide range of bitrates and better compression quality.
- MPEG-7 (2001) “Multimedia Content Description Interface”.
- MPEG-21 (2002) “Multimedia Framework”.

Three parts to MPEG

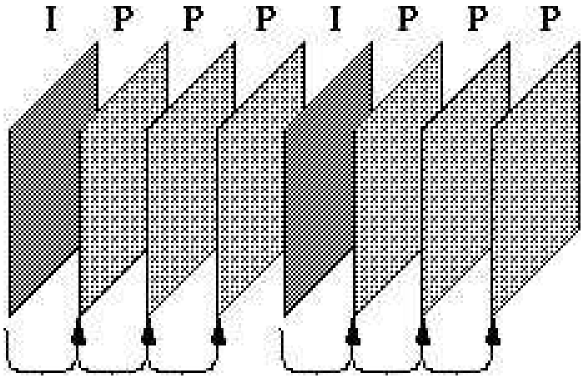
The MPEG standard has three parts:

- Video: based on H.261 and JPEG.
- Audio: based on MUSICAM (Masking pattern adapted Universal Subband Integrated Coding And Multiplexing) technology.
- System: control interleaving of streams.

MPEG video

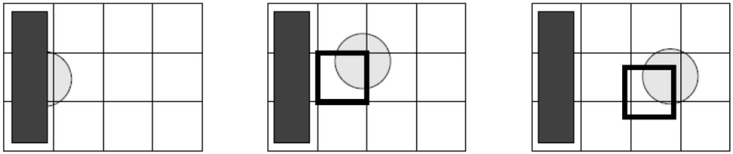
MPEG compression is essentially an attempt to overcome some shortcomings of H.261 and JPEG:

- Recall H.261 dependencies:



Bidirectional search

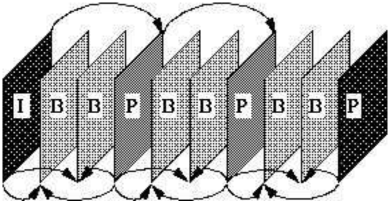
- The problem here is that many macroblocks need information that is not in the reference frame.
- For example:



- Occlusion by objects affects differencing
- Difficult to track occluded objects *etc.*
- MPEG uses forward/backward interpolated prediction.

MPEG B-frames

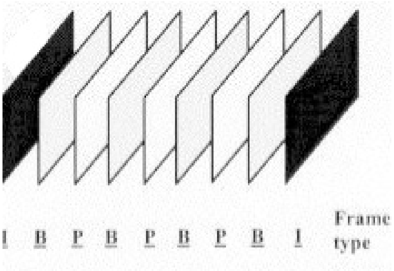
- The MPEG solution is to add a third frame type which is a bidirectional frame, or B-frame.
- B-frames search for macroblock in past and future frames.
- Typical pattern is IBBPBBPBB IBBPBBPBB IBBPBBPBB. Actual pattern is up to encoder, and need not be regular.



Example: I, P, and B frames

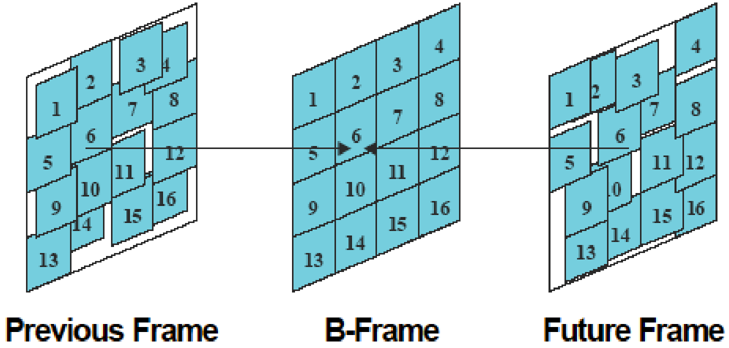
Consider a group of pictures that lasts for 6 frames:

- Given: I,B,P,B,P,B,I,B,P,B,P,B,...



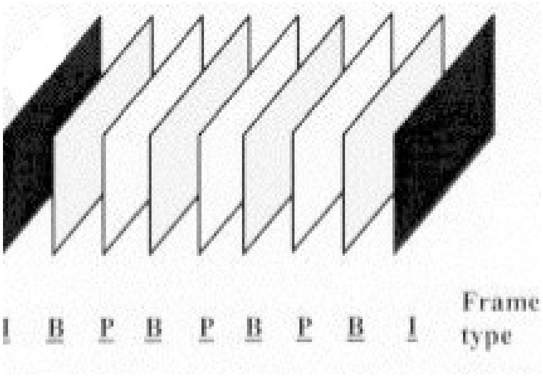
- I frames are coded spatially only (as before in H.261).
- P frames are forward predicted based on previous I and P frames (as before in H.261).
- B frames are coded based on a forward prediction from a previous I or P frame, as well as a backward prediction from a succeeding I or P frame.

Bidirectional prediction

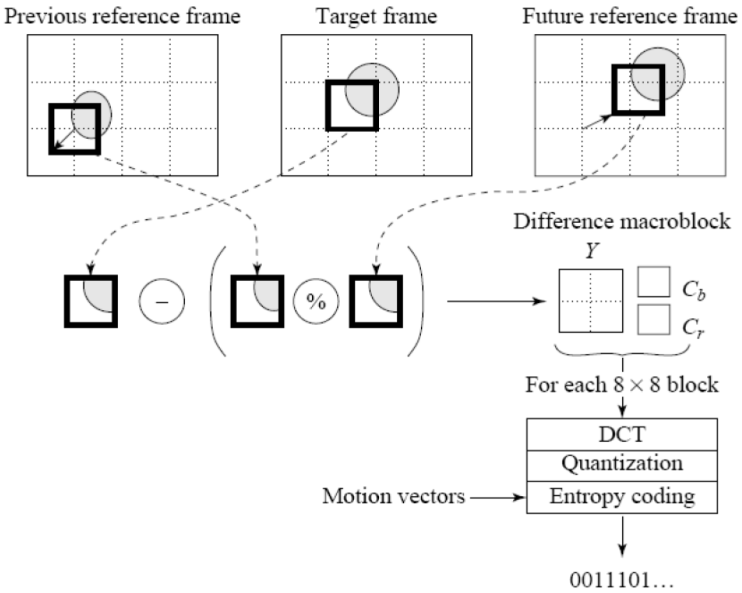


Example: I, P, and B frames

- 1st B frame is predicted from the 1st I frame and 1st P frame.
- 2nd B frame is predicted from the 1st and 2nd P frames.
- 3rd B frame is predicted from the 2nd and 3rd P frames.
- 4th B frame is predicted from the 3rd P frame and the 1st I frame of the **next** group of pictures.



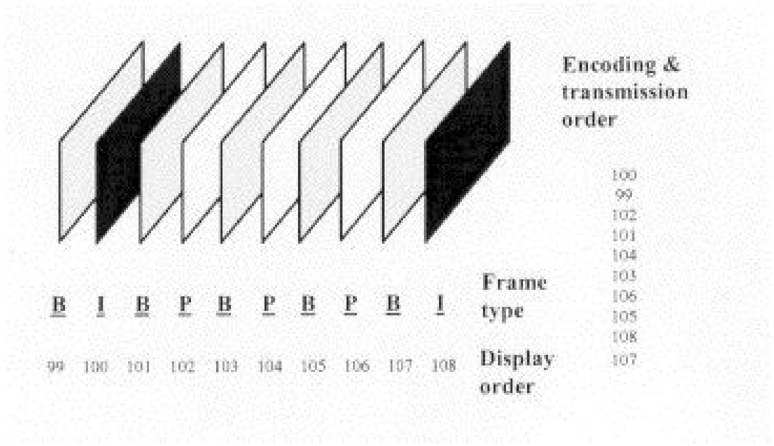
Bidirectional prediction



Backward prediction implications

Note: Backward prediction requires that the future frames that are to be used for backward prediction be encoded and transmitted first, *i.e.* out of order.

This process is summarised:



Backward prediction implications

- No defined limit to the number of consecutive B frames that may be used in a group of pictures.
- Optimal number is application dependent.
- Most broadcast quality applications, however, have tended to use 2 consecutive B frames (I,B,B,P,B,B,P,. . .) as the ideal trade-off between compression efficiency and video quality.
- MPEG suggests some standard groupings.

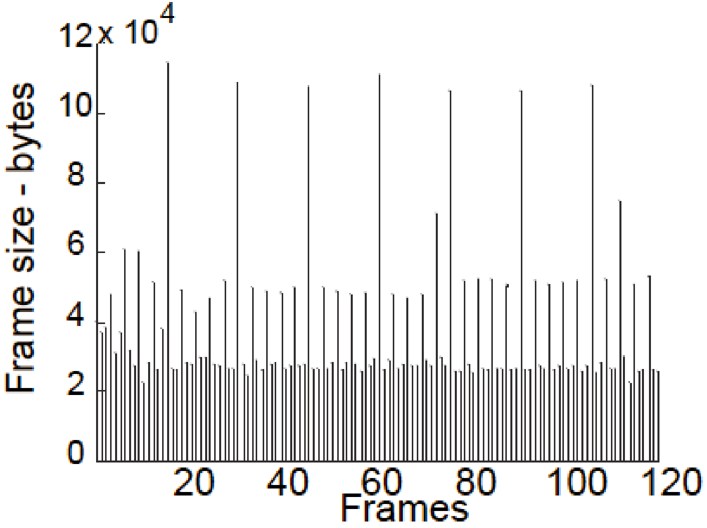
Advantage of using B-frames

- Coding efficiency.
- Most B frames use fewer bits.
- Quality can also be improved in the case of moving objects that reveal hidden areas within a video sequence.
- Better error propagation: B frames are not used to predict future frames, errors generated will not be propagated further within the sequence.

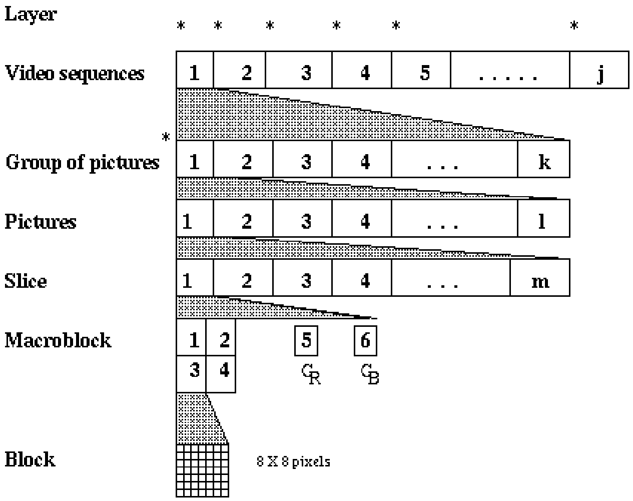
Disadvantage:

- Frame reconstruction memory buffers within the encoder and decoder must be doubled in size to accommodate the 2 anchor frames.
- More delays in real-time applications.

Frame Sizes



Random Access Points



* = possible random−access entry points

MPEG-2, MPEG-3, and MPEG-4

- MPEG-2 differences from MPEG-1
    1. Search on fields, not just frames.
    2. 4:2:2 and 4:4:4 macroblocks
    3. Frame sizes as large as 16383 x 16383
    4. Scalable modes: Temporal, Progressive,...
    5. Non-linear macroblock quantization factor
    6. A bunch of minor fixes
- MPEG-3: Originally for HDTV (1920 x 1080), got folded into MPEG-2
- MPEG-4: very low bit-rate communication (4.8 to 64 kb/sec). Around objects, not frames.