SE3IA11/SEMIP12
Image Analysis

Image Compression – Revisited

Lecturer:
Prof. James Ferryman
Computational Vision Group
Email: j.m.ferryman@reading.ac.uk

Introduction

- In the first lecture on image compression we examined the general process and methods utilised for image data reduction

- This lecture continues the theme to:
  - Compare image compression methods
  - Examine image sequence coding (video compression)

Comparison of Compression Methods

- *Transform-based* methods
  - better preserve subjective image quality
  - less sensitive to statistical image property changes

- *Prediction* methods
  - can achieve larger compression ratios in a much less expensive way
  - tend to be much faster than transform-based or vector quantization compression schemes
  - easily realised in hardware

Comparison of Compression Methods

- If compressed images are transmitted, an important property is *insensitivity* to transmission channel noise

- <u>Transform-based techniques</u> are significantly less sensitive to channel noise
  - if transform coefficient is corrupted during transmission, resulting image distortion is homogeneously spread through the image and is not too detrimental

Comparison of Compression Methods

- Erroneous transmission of a difference value in prediction compression causes not only an error in a particular pixel, it influences values in the neighbourhood
  - the predictor involved has a considerable visual effect in a reconstructed image

- Pyramid based schemes have a natural compression ability and show potential for further improvement of compression ratios
  - suitable for dynamic image compression and for progressive and smart transmission approaches

Introduction to Video Compression

- Motivation:
  - Raw video contains an immense amount of data
  - Communication and storage capabilities are limited
- Example HTDV video signal:
  - 720 x 1820 pixels/frame, progressive scanning at 60 frames/s

$$\left(\frac{720\ x1280\ pixels}{frame}\right)\left(\frac{60\ frames}{\sec}\right)\left(\frac{3\,colours}{pixel}\right)\left(\frac{8\,bits}{colour}\right)=1.3Gb/s$$

  - 20 Mb/s HDTV channel bandwidth
  - Requires compression by factor of ~ 70

Introduction to Video Compression

- Video is a sequence of frames (images) that are *related*
- Related along the temporal dimension
  - therefore, temporal redundancy exists
- Main addition over image compression studied previously
  - temporal redundancy
  - video coder must exploit the *temporal redundancy*

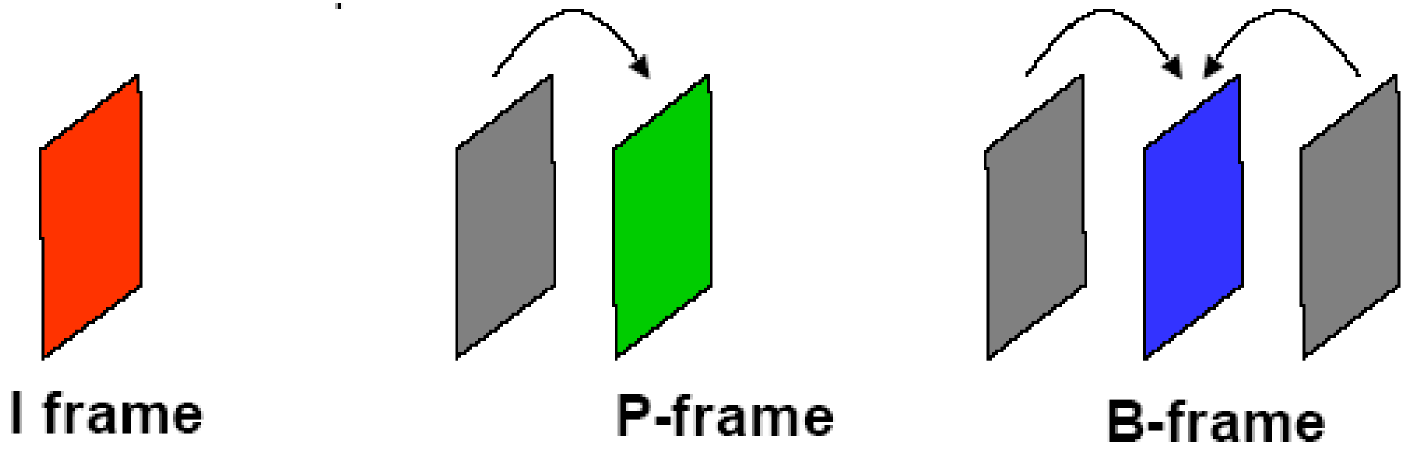
Temporal Processing

- Usually there exists a high frame rate
  - Significant temporal redundancy
- Possible representations along the temporal dimension include:
  - Transform methods
    - Good for constant velocity uniform global motion
    - Inefficient for non-uniform motion
      - real-world motion
    - Requires a large number of frame stores
      - leads to delay (+ memory cost)
  - Predictive methods
    - Good performance using only 2 frame stores
    - However, simple frame differencing is insufficient

Video Compression

- Main advantage over image compression:
  - Exploit the temporal redundancy
- *Predict current frame* based on previously coded frames
- 3 types of coded frames:
  - I-frame: Intra-coded frame, coded independently of all other frames
  - P-frame: Predictively coded frame, coded based on previously coded frame
  - B-frame: Bi-directionally predicted frame, coded based on both previous and future coded frames

Coded Frames



Temporal Processing

Motion-Compensated Prediction

- Simple frame differencing *fails* when there is motion
- Motion must be taken into account
  - Motion-compensated (MC) prediction
- MC prediction generally provides significant improvements
- However:
  - how can we estimate motion?
  - how can we form MC-prediction?

Temporal Processing: Motion Estimation

- The ideal situation is to:
  - Partition the video into moving objects
  - Describe the object motion

  In general, this is a very difficult task

- Practical approach: *block-matching motion estimation*
  - Partition each frame into blocks
  - Describe the motion of each block
  - Requires no object identification

  In general, good robust performance

Block Matching Motion Estimation

- Assumptions:
  - Translational motion within block:

    $$f(n_1,n_2,k_{cur}) = f(n_1\text{-}m_{v1}, n_2\text{-}m_{v2}, k_{ref})$$

  - All pixels within each block have the same motion

- ME Algorithm:
  - Divide current frame into non-overlapping N_1 x N_2 blocks
  - For each block, find the best matching block in reference frame
- MC-Prediction Algorithm:
  - Use best matching blocks of reference frame as prediction of blocks in current frame

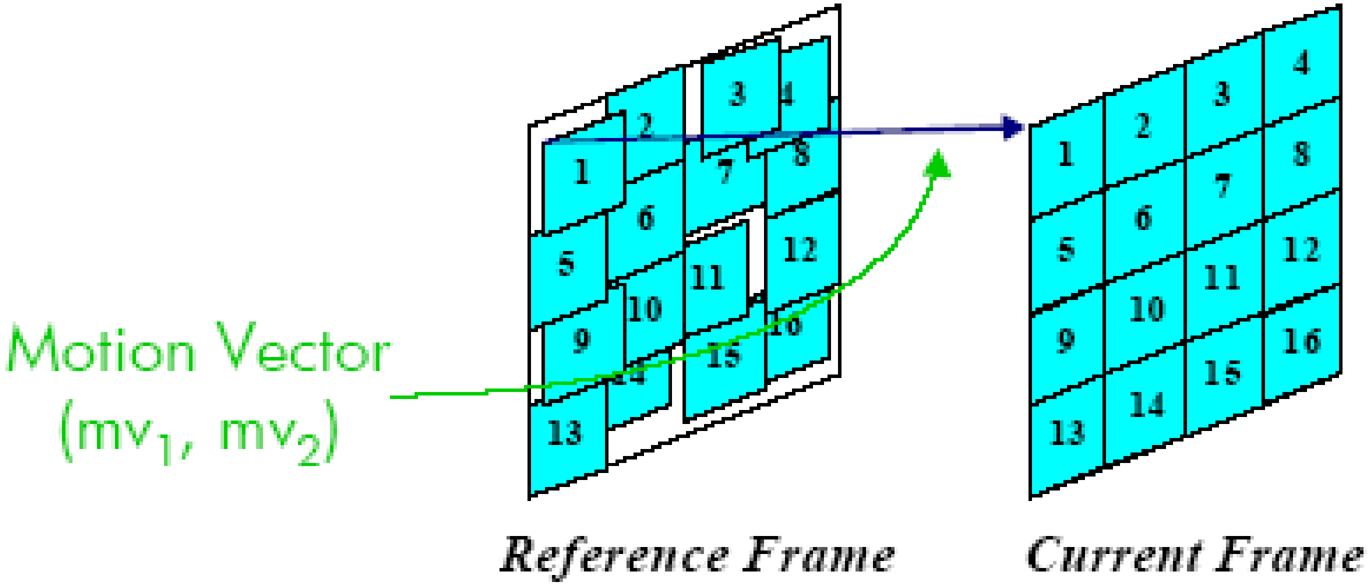
Block Matching Motion Estimation

- For each block in the current frame we need to search for best matching block in the reference frame
- A number of metrics exist for the "best match", e.g.:

$$MSE = \sum_{(n_1,n_2)\in Block}\sum [f(n_1,n_2,k_{cur}) - f(n_1 - mv_1, n_2 - mv_2, k_{ref})]^2$$

- Candidate blocks include all blocks in e.g. (±32, ±32) pixel area
- Strategies for searching candidate blocks for best match include:
  - Full search: examine all candidate blocks
  - Partial (fast) search: examine a carefully selected subset
- Estimation of motion for best matching block: "motion vector"

Block Matching Motion Estimation



Motion Vectors and Motion Vector Field

- Motion vector
  - Expresses the relative horizontal and vertical offsets (mv_1, mv_2), or motion, of a given block from one frame to another
  - Each block has its own motion vector

- Motion vector field
  - Collection of motion vectors for all the blocks in a frame

Block Matching: Summary

- The main issues to consider are:
  - Block size
  - Search range
  - Motion vector accuracy
- Motion typically estimated only from raw intensities
- Advantages:
  - Good, robust performance for compression
  - Resulting motion vector field is easy to represent (one MV per block) and useful for compression
  - Simple, periodic structure, easy hardware implementations
- Disadvantages:
  - Assumes translational motion model
    - Breaks down for more complex motion
  - Often produces blocking artifacts

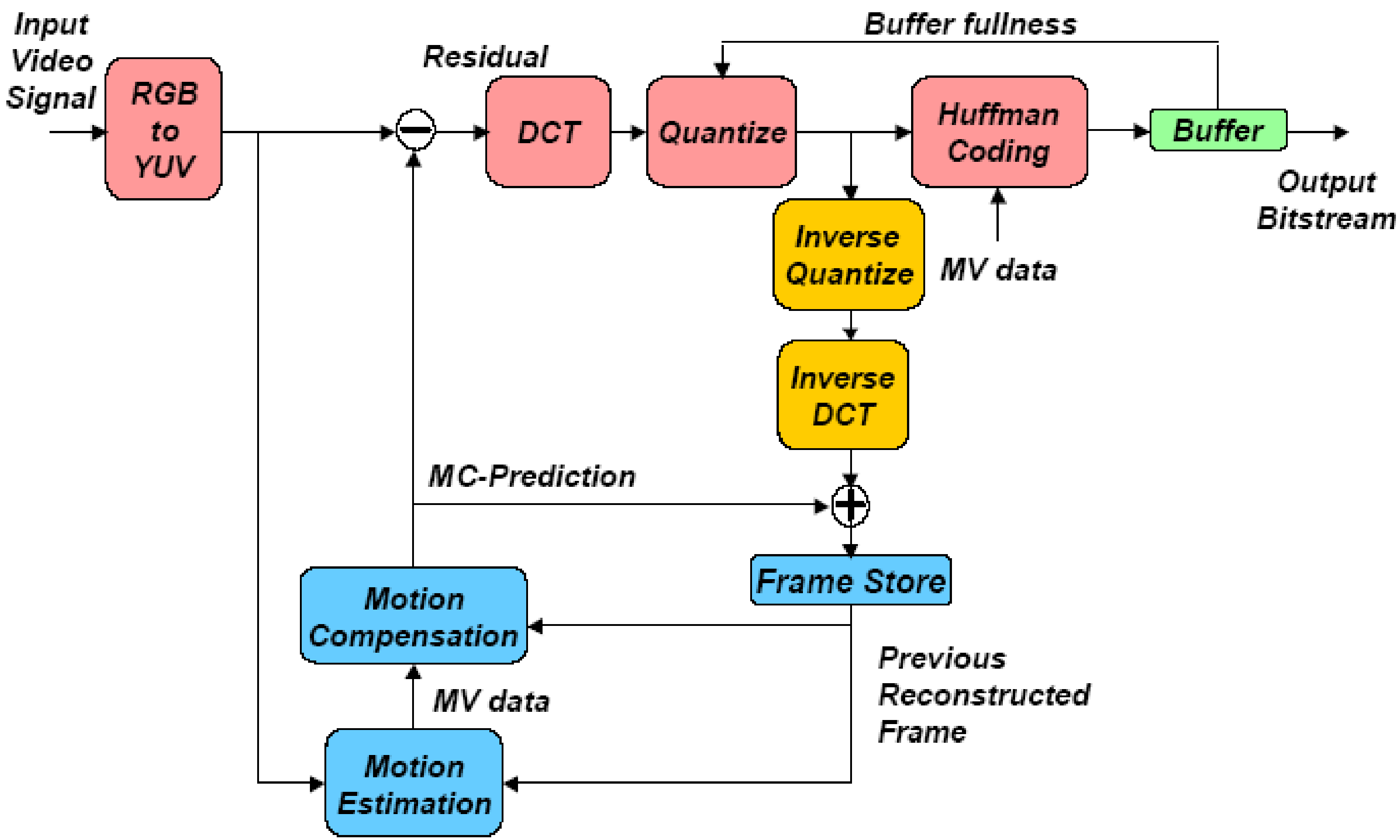
Summary of Temporal Processing

- Use MC-prediction (P and B frames) to reduce temporal redundancy
- MC-prediction usually performs well
- MC-prediction provides:
  - Motion vectors
  - MC-prediction error or residual
    - Code error with conventional image coder
- Occasionally MC-prediction may perform badly
  - E.g.: complex motion, new imagery (occlusions)
    - Approach:
      - Identify blocks where prediction fails
      - Code block *without* prediction

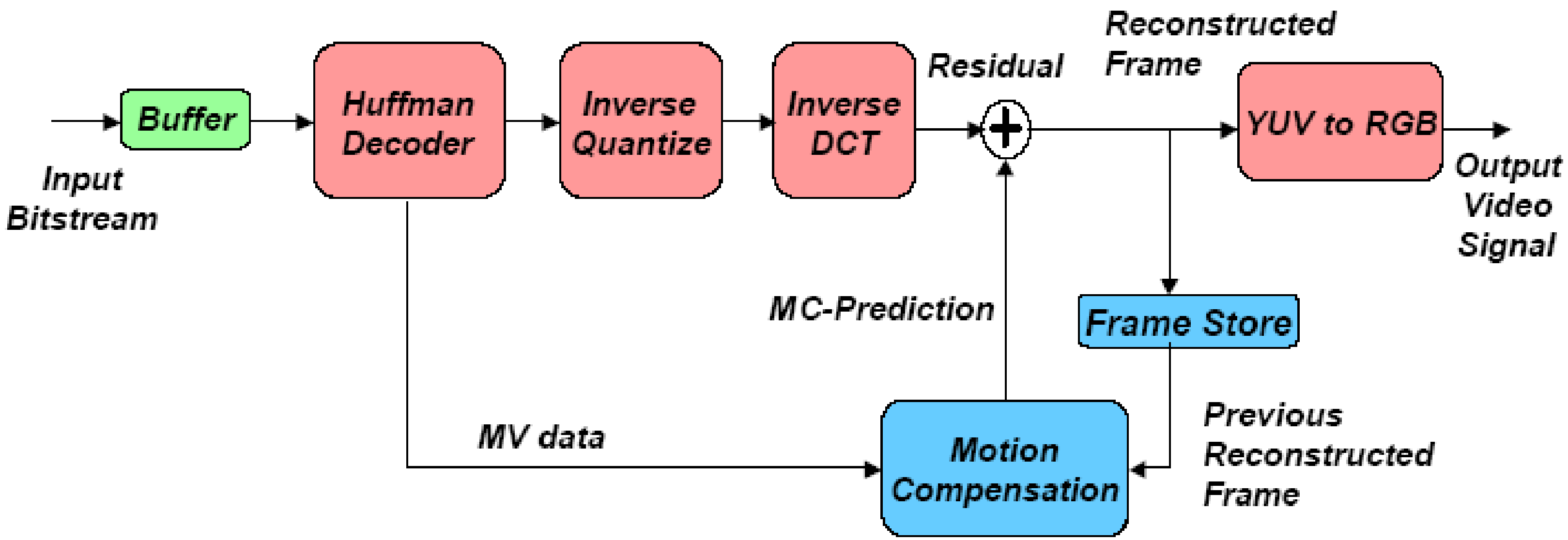
Video Compression Architecture

- Exploits the redundancies that we have learnt:
  - Temporal: MC-prediction (P and B frames)
  - Spatial: Block DCT
  - Colour: Colour space conversion
- Scalar quantization of DCT coefficients
- Zigzag scanning, run length & Huffman coding of the non-zero quantized DCT coefficients

Example Video Encoder



Example Video Decoder



Acknowledgements

- HP – Image and Video Coding

Summary

- We have examined some additional aspects of image compression
- The next lecture today covers symbolic feature extraction