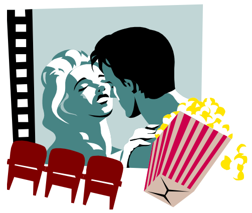
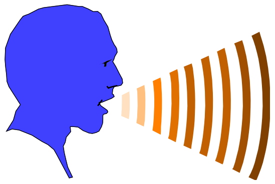



Image and Video Processing

Block-Based Hybrid Video Coding

Yao Wang
Tandon School of Engineering, New York University

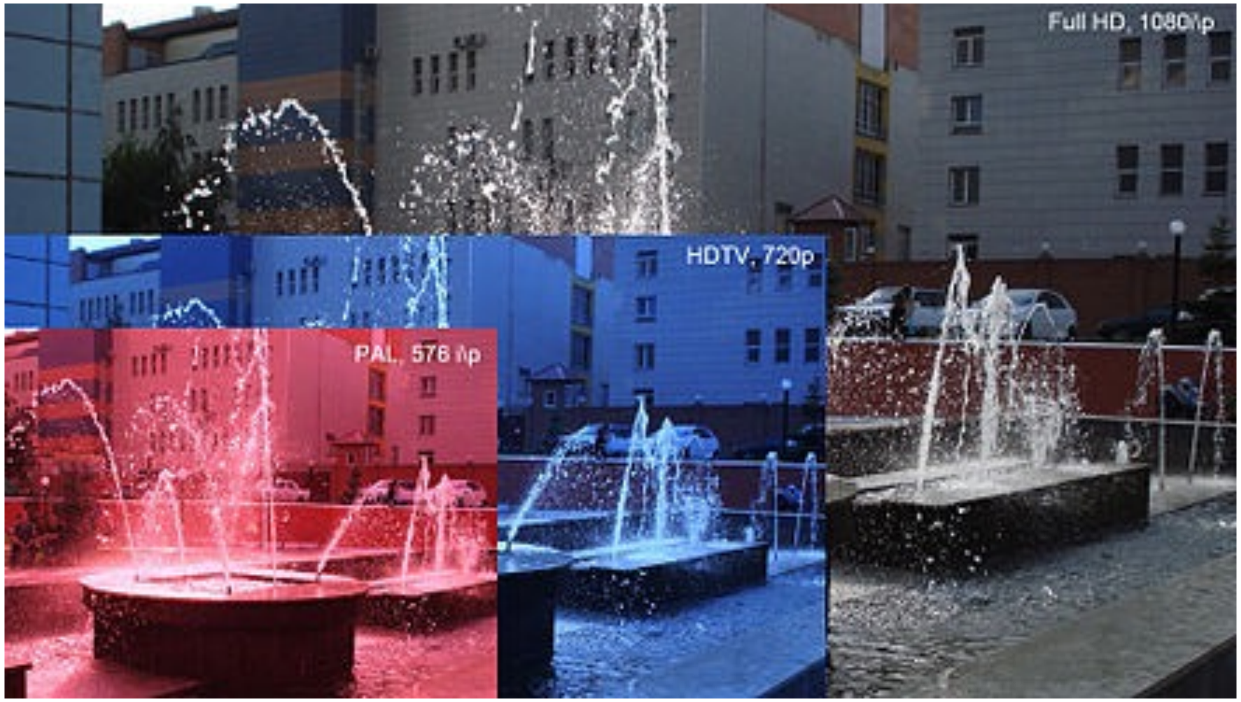
Outline

- Temporal prediction (inter-prediction)
  - Uni-directional, bi-directional prediction
- Spatial prediction (intra-prediction)
- Transform coding of prediction error and coding of side information
- Rate-distortion optimization (RDO) for motion estimation and coding mode selection
- Rate control, Deblocking filtering
- GoP Structure
- Deep learning for image and video coding

Why compression?

Image / Video format	**Size**
One small VGA size picture (640x480, 24-bit color)	922 KB
One large 12 MB pixel picture (3072x4096) 24-bit color still image	36 MB
Animation (320x640 pixels, 16-bit color, 16 frame/s)	6.25 MB/second
SD Video (720x480 pixels, 24-bit color, 30 frame/s)	29.7 MB/second
HD Video (1920x1080 pixels, 24-bit color, 60 frame/s)	356 MB/second

360 video ?
HDR video ?
AR/VR applications?



Video codec is ubiquitous!

- In your camera, phone, computer browser!
- You use it every day
  - Video call (FaceTime, Skype, WeChat, …)
  - On-demand video streaming (Youtube, Netflix, …)
  - Live streaming of popular events (sports, concerts, …)
  - Our online class (zoom, …)

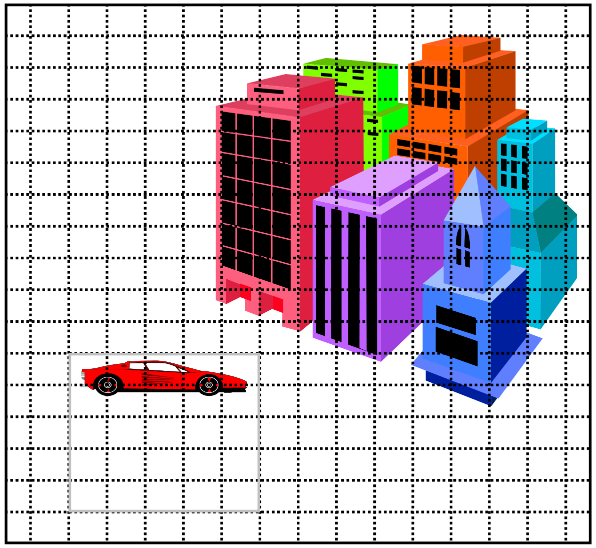
Predictive Coding for Image and Video

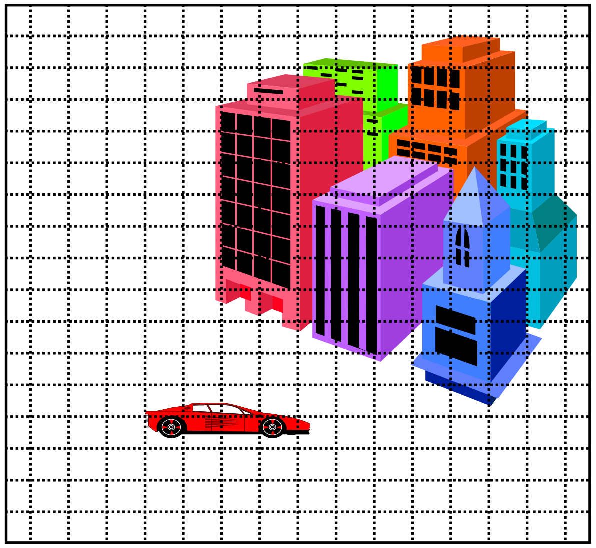
- For images: we can predict a current sample from previously coded samples.
  - In JPEG: predictive coding is used to code the DC coefficient of a block, which is the mean of the block. The current block DC is predicted from the previous block DC.
  - We can also use non-linear adaptive predictor
- For video: we apply prediction both among pixels in the same frame (intra-prediction or spatial prediction), and also among pixels in adjacent frames (inter-prediction or temporal prediction)
  - Both spatial and temporal predictor are adaptive (hence non-linear!)
  - Prediction error is coded using transform coding.

Outline

➡
- Temporal prediction (inter-prediction)
  - Uni-directional, bi-directional prediction
- Spatial prediction (intra-prediction)
- Transform coding of prediction error and coding of side information
- Rate-distortion optimization (RDO) for motion estimation and coding mode selection
- Rate control, Deblocking filtering
- GoP Structure
- Deep learning for image and video coding

Characteristics of Typical Videos

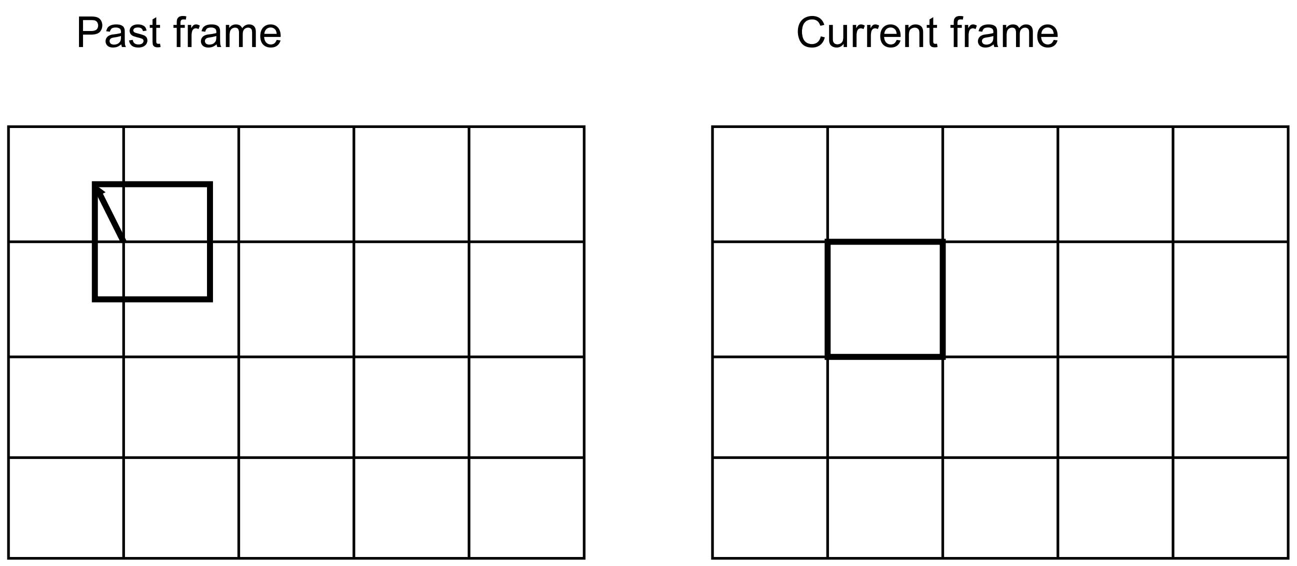

Frame *t-1*


Frame *t*

Adjacent frames are similar and changes are due to object or camera motion

We can predict most blocks in Frame t from Frame t-1 accurately, and send only prediction error!

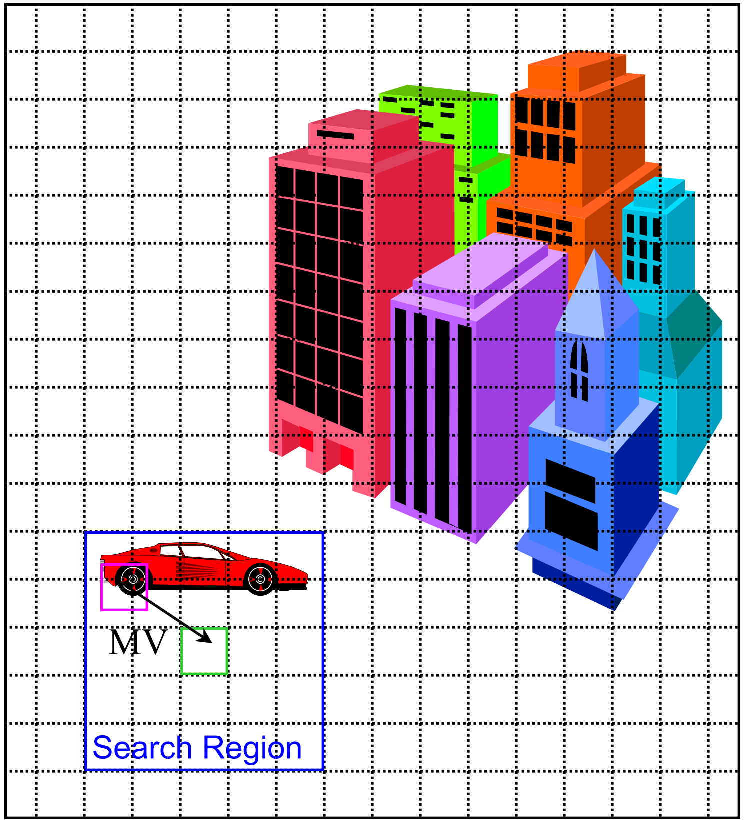
Block-Based Motion Compensated Prediction



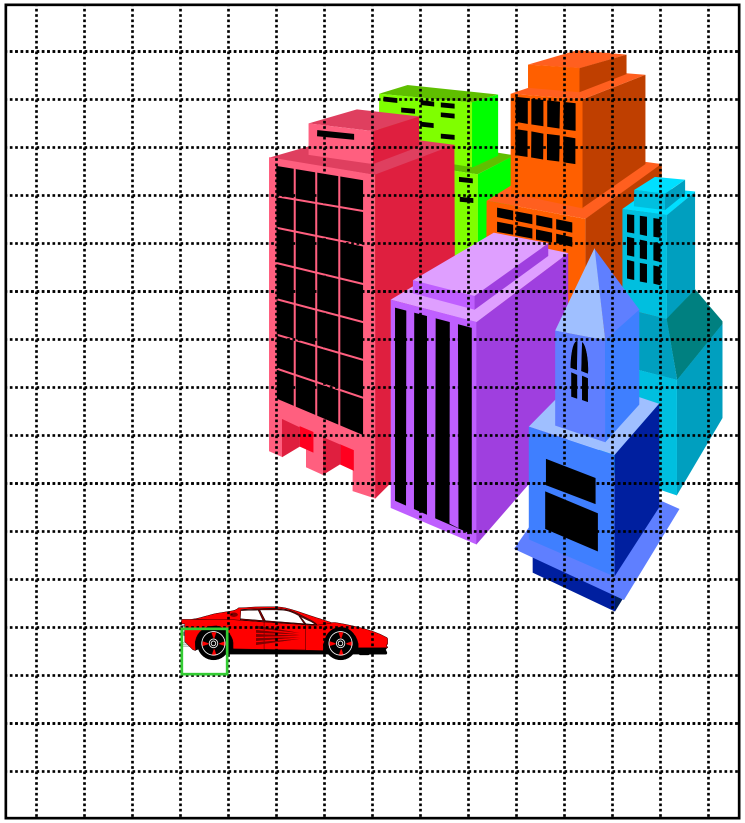
- Divide the current frame into blocks
- Find the best matching block in the previous frame
  - Using EBMA, integer or fractional pel accuracy
- Code the prediction error and the motion vector
- Big win: Improves compression by factor of 5-10

From Amy Reibman

Block Matching Algorithm for Motion Estimation



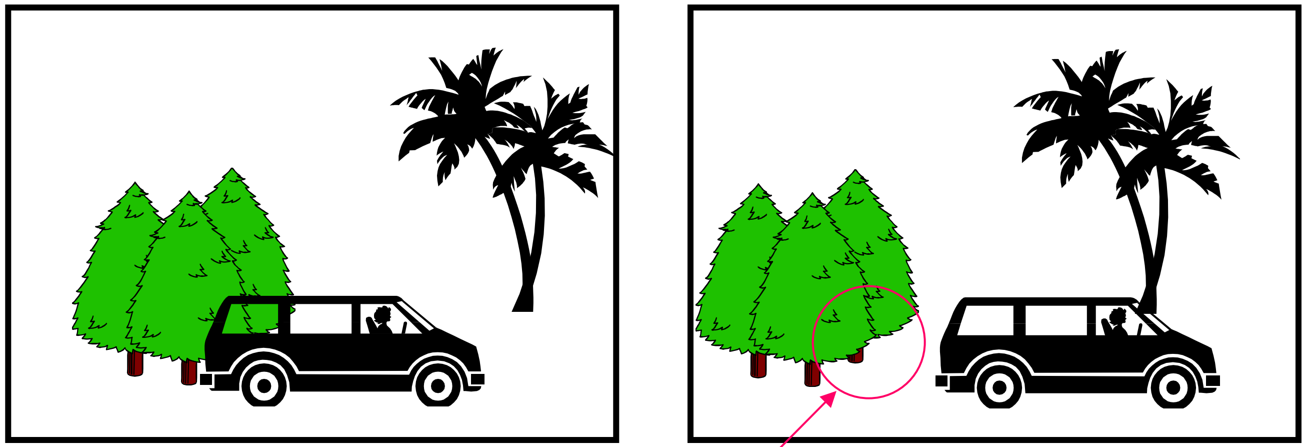
Reference Frame, target frame



Predicted frame, anchor frame

Reference frame can be before or after the predicted frame in the original frame order!
Reference frame must be coded before the predicted frame!
Fractional-pel step-size MV search is often used to yield more accurate prediction

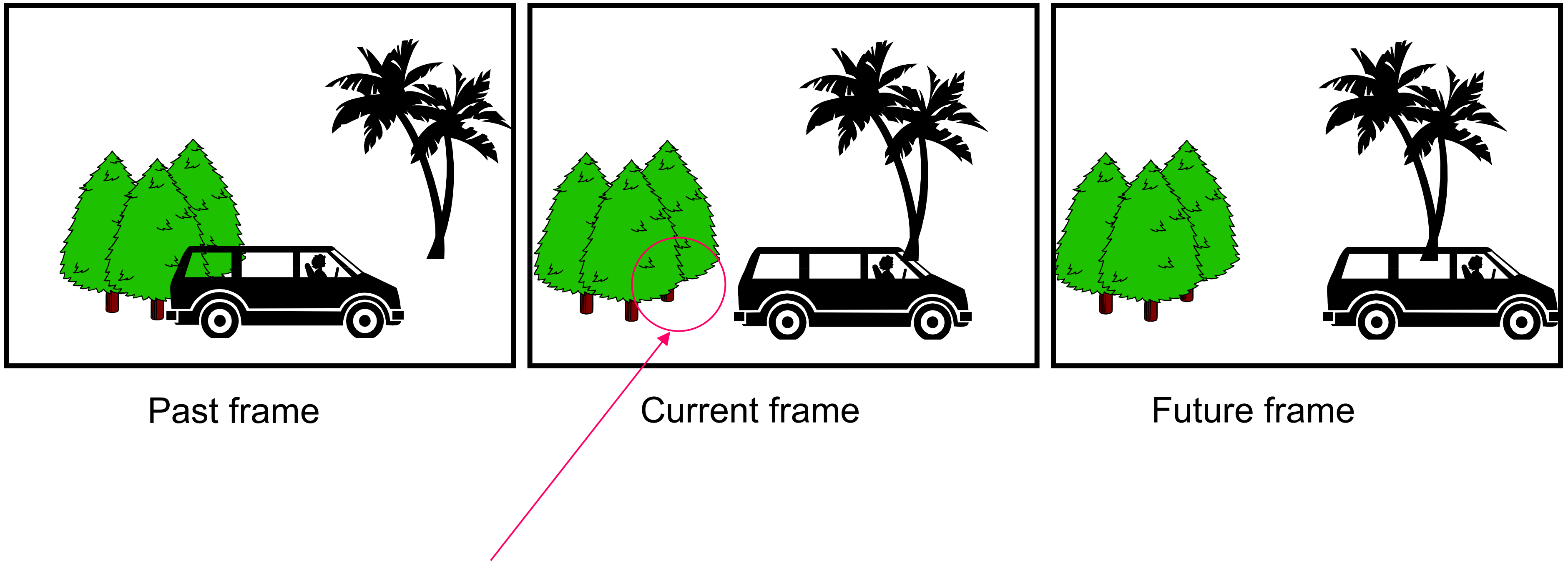
Problems for Uni-Directional Temporal Prediction



All objects **except** this area have already been sent to decoder in "past frame"

From Amy Reibman

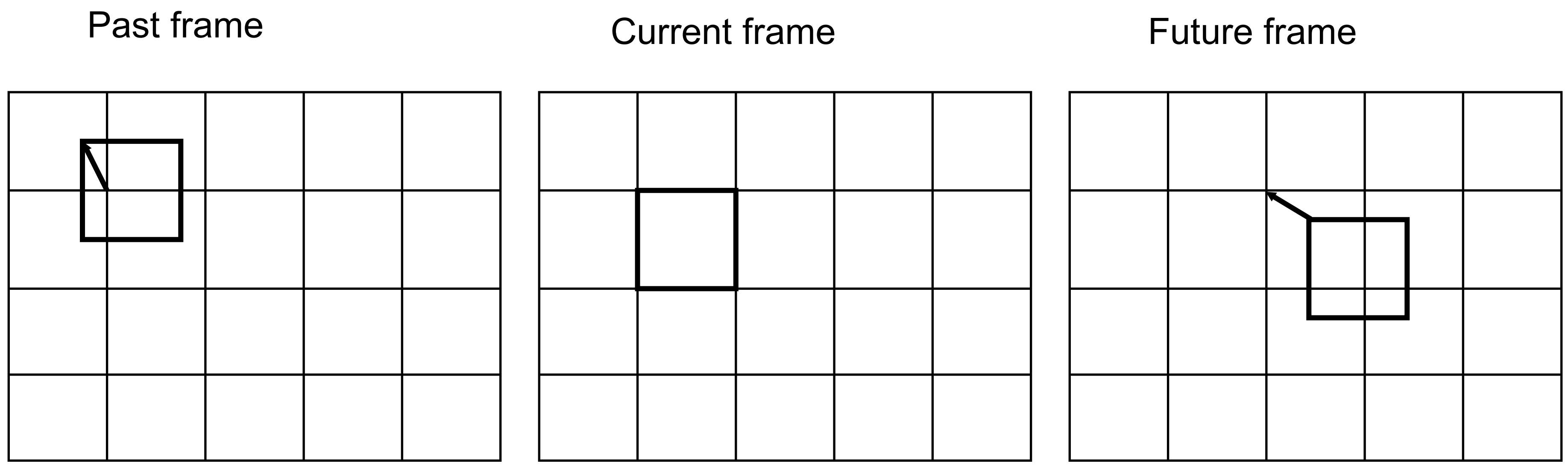
Bi-directional Prediction



This area can now be predicted using "future frame"

From Amy Reibman

Block-Based Bidirectional Prediction



- Code past frame and future frame first, future predicted from past
- Then code the current frame using bi-directional prediction
  - Divide the current frame into blocks
  - For each block, find best matching blocks in both future and past frames
  - Using a weighted average of both matching blocks as the prediction
  - Code the prediction error block and Two motion vectors
- Helps when there is occlusion or uncovered objects

Motion-Compensated Temporal Prediction (summary)

- No Motion Compensation (assume zero motion):
  - Work well in stationary regions

$$\hat{f}(t,m,n) = f(t-1,m,n)$$

- Uni-directional Motion Compensation:
  - Does not work well for uncovered regions by object motion

$$\hat{f}(t,m,n) = f(t-1,m-d_x,n-d_y)$$

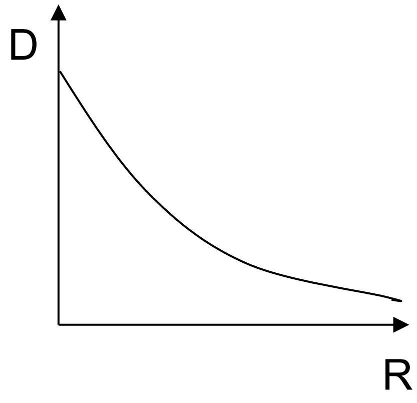
- Bi-directional Motion Compensation
  - Can handle better uncovered regions

$$\hat{f}(t,m,n) = w_b f(t-1,m-d_{b,x},n-d_{b,y}) \\ + w_f f(t+1,m-d_{f,x},n-d_{f,y})$$

Rate-Distortion Basics

- Performance metric for evaluating a compression scheme:
  - R-D curve, Relation between distortion (in MSE) and bit rate
- Scalar quantization:
  - Quantize one variable by itself
- RD Bound for optimal scalar quantization at high rate (high rate approximation)

$$\sigma_q^2 = \varepsilon^2 \sigma_s^2 2^{-2R}$$



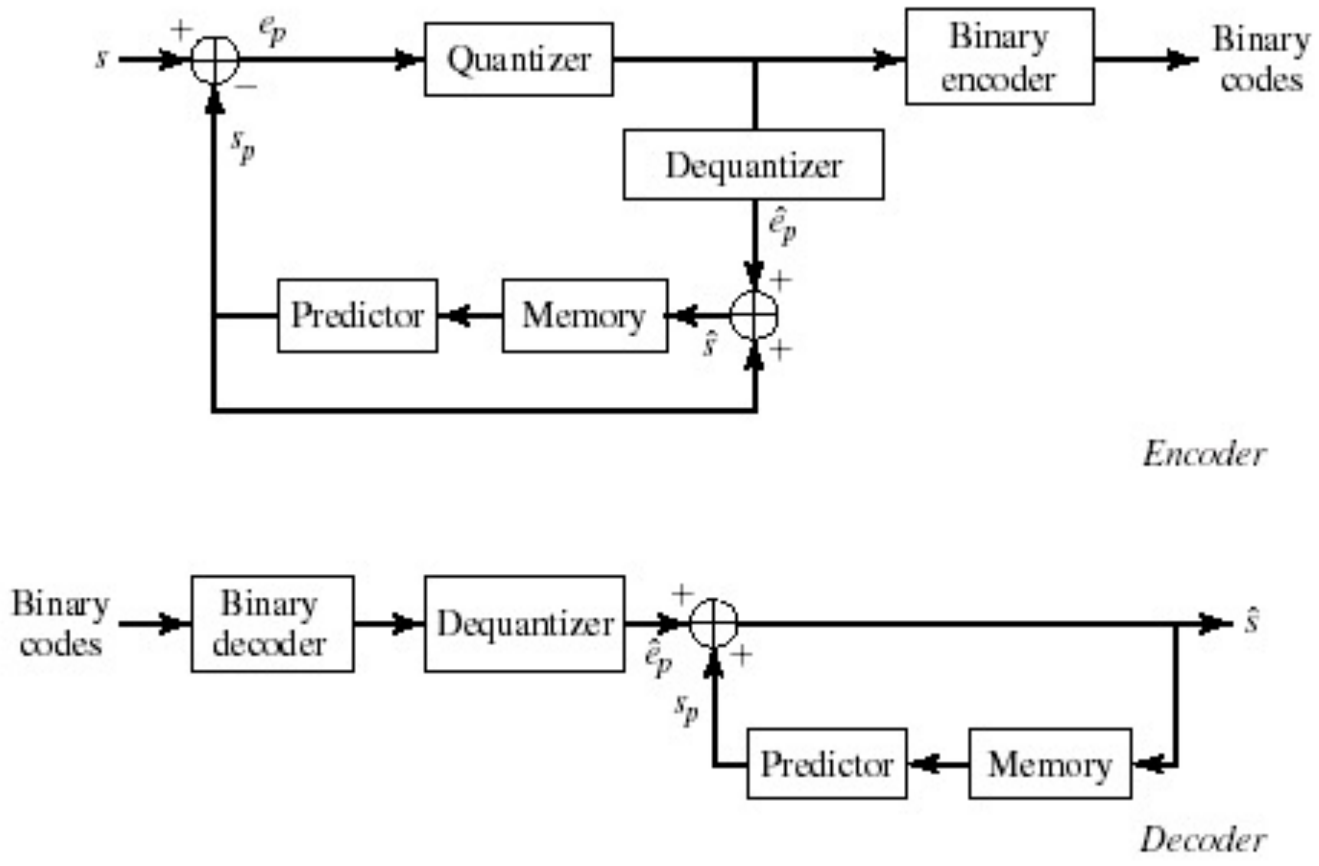
  - σ_s^2 is the variance of the signal
  - σ_q^2 is linearly proportional to signal variance σ_s^2
  - σ_q^2 decays exponentially with bit rate R
  - ε^2 depends on the distribution of the signal

Why predictive coding?

- The rate-distortion performance depends on the variance of the signal to be coded. The variance of the prediction error is significantly smaller than the variance of the original signal
- Goal of prediction design:
  - Minimize the prediction error variance! (=mean square error of prediction)
- Closed-loop prediction:
  - Predict from decoded pixels (with quantization error)
  - Quantization error for the prediction error = quantization error for the original signal

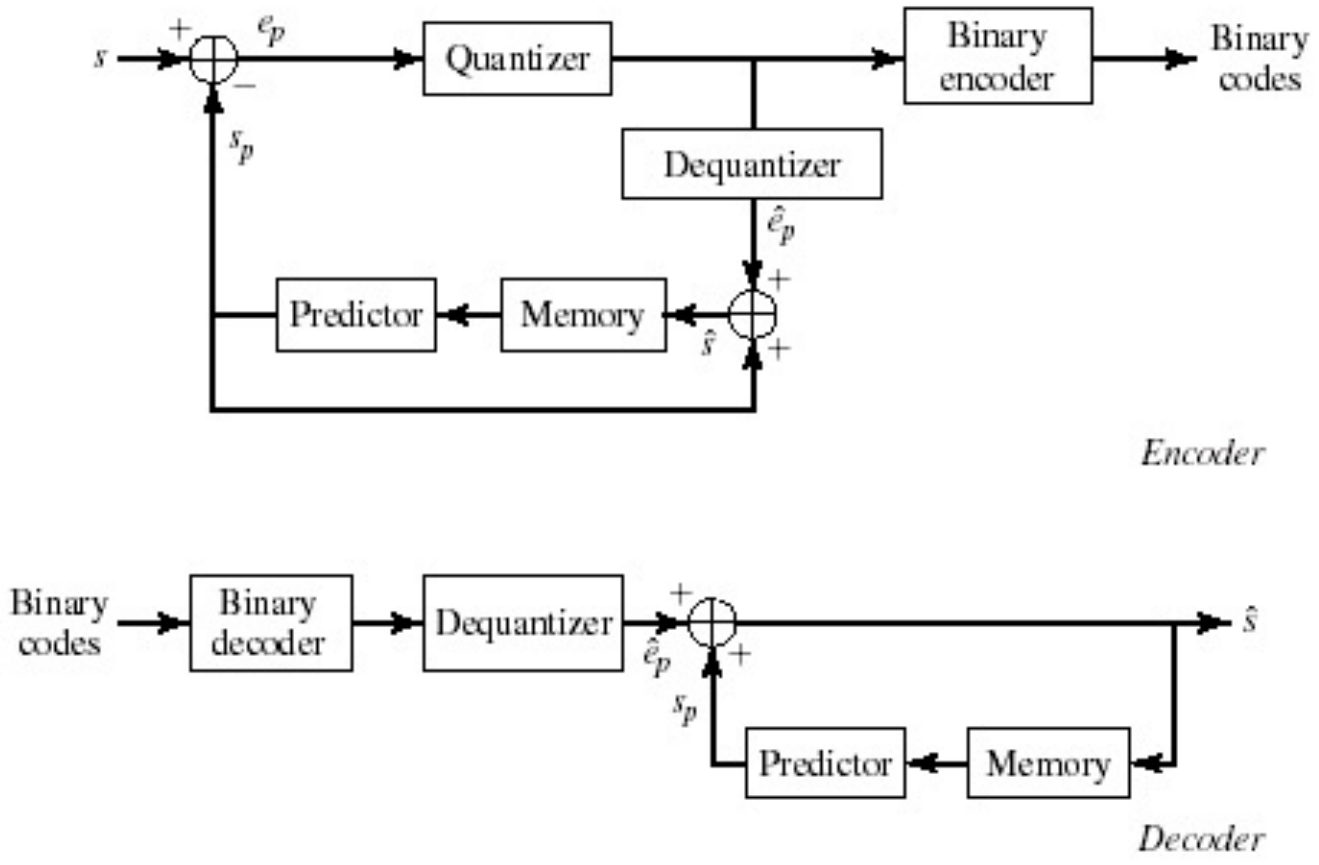
Encoder and Decoder Block Diagram

- Prediction is based on previously decoded (with quantization error) samples
  - Encoder and decode can produce the same prediction!
- Closed-loop prediction to avoid encoder/decoder mismatch!



Distortion in Predictive Coder

- Reconstruction error = quantization error for the prediction error



$$e_p = s - s_p$$

$$\hat{e}_p = e_p + e_q$$

$$\hat{s} = s_p + \hat{e}_p$$

$$= s - e_p + e_p + e_q$$

$$= s + e_q$$

$$e = \hat{s} - s = e_q!$$

Gain of Uni-Directional Prediction

- Uni-directional motion compensated prediction

$$\hat{f}(t,m,n) = f(t-1, m-d_x, n-d_y)$$

Assume the signal is stationary with variance σ_p^2 and

the correlation coefficient between $f(t,m,n)$ and $f(t\text{-}1,m\text{-}d_x,n-d_y)$ is ρ

$$\begin{aligned}\sigma_p^2 &= E\left\{\left(f(t,m,n) - f(t\text{-}1,m\text{-}d_x,n-d_y)\right)^2\right\} \\ &= E\left\{f^2(t,m,n) + f^2(t\text{-}1,m\text{-}d_x,n-d_y) - 2f(t,m,n)f(t\text{-}1,m\text{-}d_x,n-d_y)\right\} \\ &= 2\sigma_s^2(1-\rho)\end{aligned}$$

Gain over coding a pixel directly $G_p = \dfrac{\sigma_s^2}{\sigma_p^2} = \dfrac{1}{2(1-\rho)}$

Reduction in quantization error under the same bit rate

ρ is typically in the range of (0.9, 1)

Assuming $\rho = 0.9, G = 5, => 10log_{10}$5=7dB gain in PSNR at same bit rate!

Gain of Bi-Directional Prediction

- Bi-directional motion compensated prediction

$$\hat{f}(t,m,n) = w_b f(t-1, m-d_{b,x}, n-d_{b,y}) + w_f f(t+1, m-d_{f,x}, n-d_{f,y})$$

Assume the signal is stationary with variance σ_s^2 and the correlation coefficient between $f_0 = f(t,m,n)$ and $f_1 = (t\text{-}1, m\text{-}d_{b,x}, n-d_{b,y})$ and that between $f(t,m,n)$ and $f_2 = f(t+1, m\text{-}d_{f,x}, n-d_{f,y})$ are both ρ, the correlation coefficient between f_1 and f_2 is ρ^2. Further consider the special case of $w_b = w_f = 1/2$

$$\sigma_p^2 = E\left\{\left(f_0 - \frac{1}{2}f_1 - \frac{1}{2}f_2\right)^2\right\} = E\left\{f_0^2 + \frac{1}{4}f_1^2 + \frac{1}{4}f_2^2 - f_0 f_1 - f_0 f_2 + \frac{1}{2}f_1 f_2\right\}$$

$$= \sigma_s^2(1 + \frac{1}{4} + \frac{1}{4} - \rho - \rho + \frac{1}{2}\rho^2) = \sigma_s^2\left(\frac{3}{2} - 2\rho + \frac{1}{2}\rho^2\right) = \sigma_s^2 \frac{(1-\rho)(3-\rho)}{2}$$

Gain over coding a pixel directly $G_B = \dfrac{\sigma_s^2}{\sigma_p^2} = \dfrac{2}{(1-\rho)(3-\rho)}$

G=9.5 for $\rho = 0.9$

$\dfrac{G_B}{G_P} = \dfrac{4}{(3-\rho)}$ is close to 2 since ρ is close to 1.

=> $10log_{10}2$=3dB gain in PSNR over uni-directional prediction at same bit rate!

Multiple Reference Frame Temporal Prediction

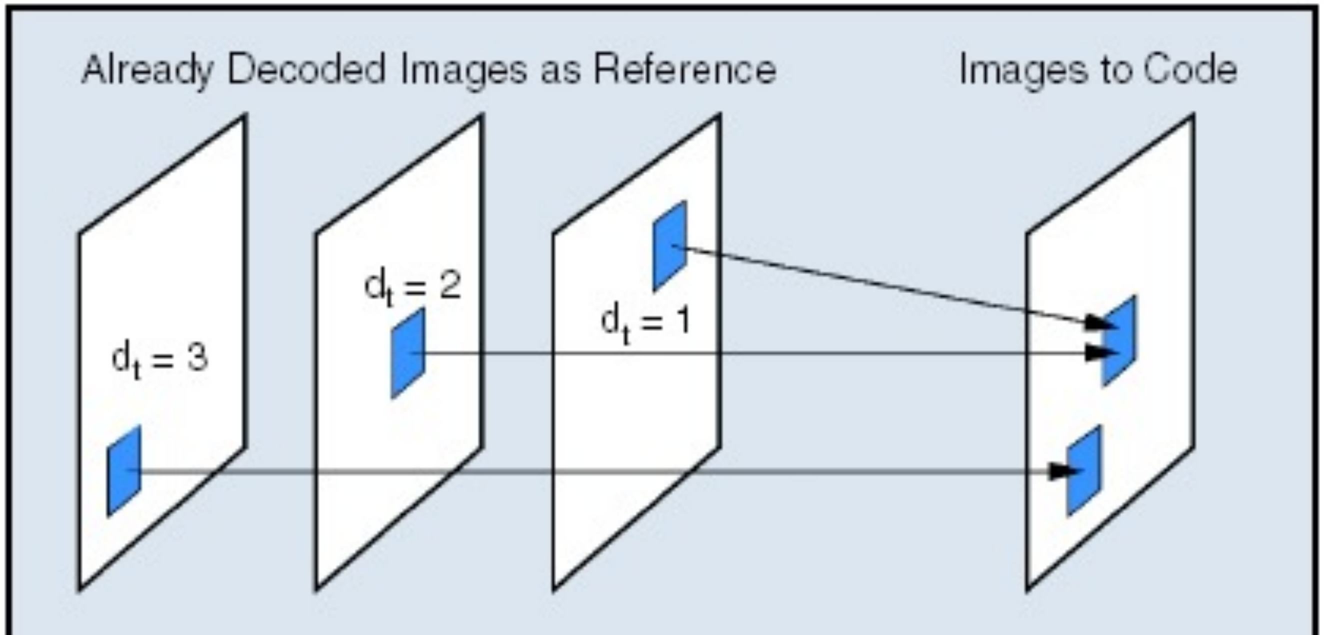


Figure 9. Motion-compensated prediction with multiple reference images. In addition to the motion vector, also an image reference parameter d_t is transmitted.

One may choose the best prediction among all frames (MV information is described by motion vector and frame index), or use a weighted average of the predictions from all reference frames.

Pop Quiz

- Why use temporal prediction?
- How to perform temporal prediction?
- Where does uni-directional prediction fail?
- What are the pros and cons of using bi-directional prediction?
- What is the problem of open-loop prediction? What is closed-loop prediction?

Pop Quiz (w/ Answers)

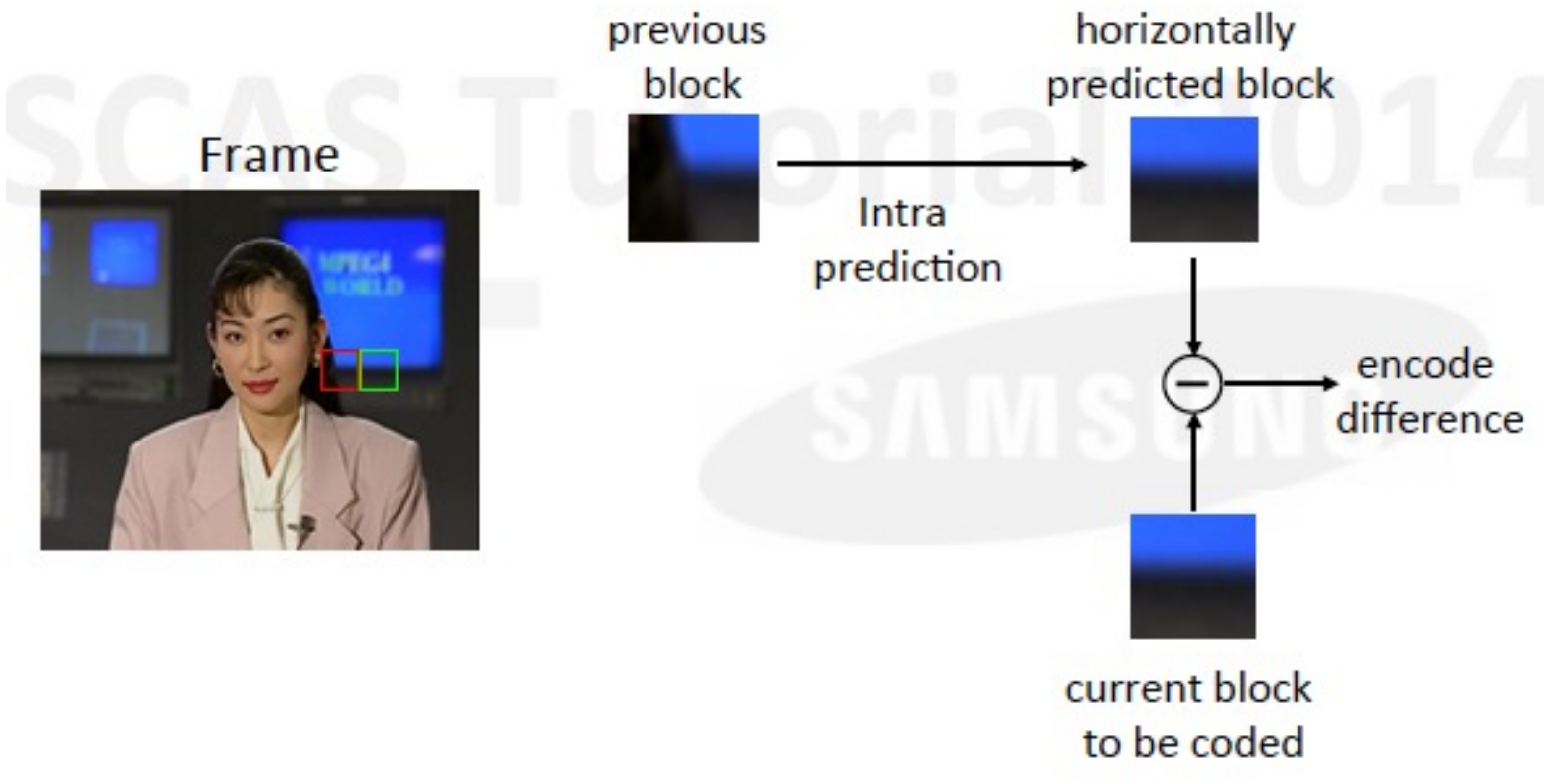
- Why use temporal prediction?
  - Intuitively: If prediction is accurate, prediction error will be zero and don’t need to be coded.
  - Theoretically: To reduce the variance and hence the bit rate of the signal to be coded
- How to perform temporal prediction?
  - No motion compensation (only work well for stationary region)
  - Predicting from the past frame (uni-directional)
  - Predicting from both the past and future frame (bi-directional)
  - Predicting from more than two frames (multiple reference frames)
- Where does uni-directional prediction fail?
  - Uncovered regions
- What are the pros and cons of using bi-directional prediction?
  - Pro: help in uncovered regions (only available in the future frame, not the past frame)
  - Con: coding delay (cannot code the current frame until the next frame is coded)
- What is the problem of open-loop prediction?
  - Open-loop: Predict from the original pixels in the previous or future frames. But decode can only predict from the decoded pixles. Encoder-decoder mismatch -> Error propagation
  - Closed-loop: Predict from the decoded pixels in the previous or future frames

Outline

- Temporal prediction (inter-prediction)
  - Uni-directional, bi-directional prediction
- Spatial prediction (intra-prediction)
- Transform coding of prediction error and coding of side information
- Rate-distortion optimization (RDO) for motion estimation and coding mode selection
- Rate control, Deblocking filtering
- GoP Structure
- Deep learning for image and video coding

Spatial Prediction

- General idea:
  - A pixel in the new block is predicted from previously coded pixels in the same frame
  - What neighbors to use?
  - What weighting coefficients to use?
- Content-adaptive prediction
  - No edges: use all neighbors
  - With edges: use neighbors along the same direction
  - The best possible prediction pattern can be chosen from a set of candidates, similar to search for best matching block for inter-prediction
    - H.264 has many possible intra-prediction pattern


Frame
previous block
Intra prediction
horizontally predicted block
encode difference
current block to be coded

H.264 Intra Prediction Modes

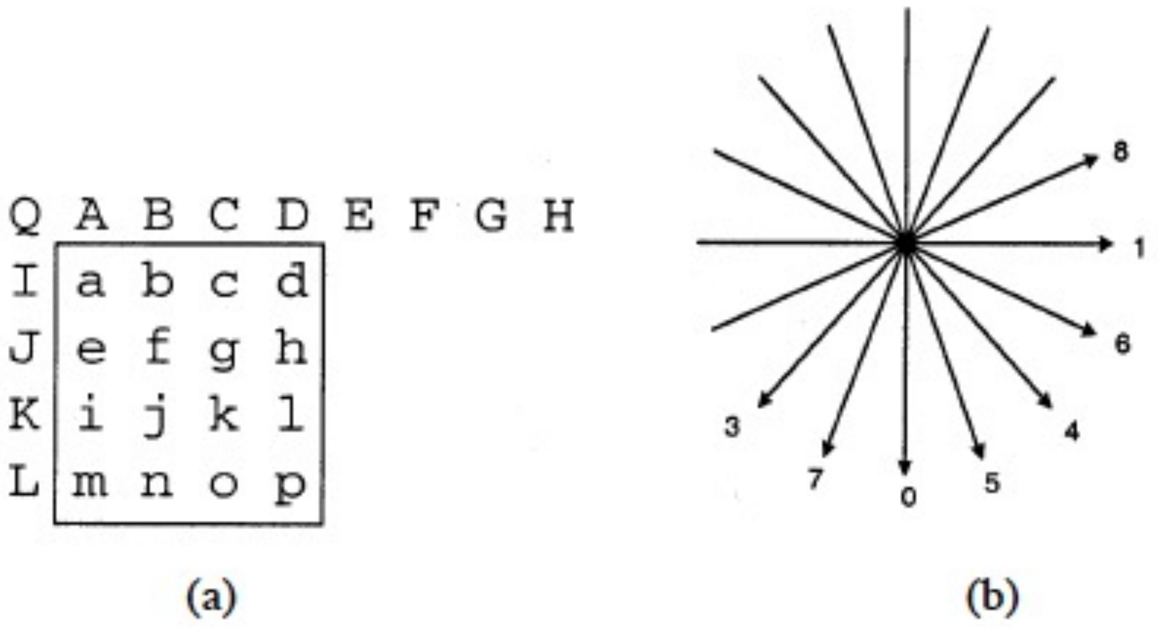


Fig. 10. (a) Intra_4×4 prediction is conducted for samples a-p of a block usin samples A-Q. (b) Eight "prediction directions" for Intra_4×4 prediction.

Intra-Prediction modes in H.264

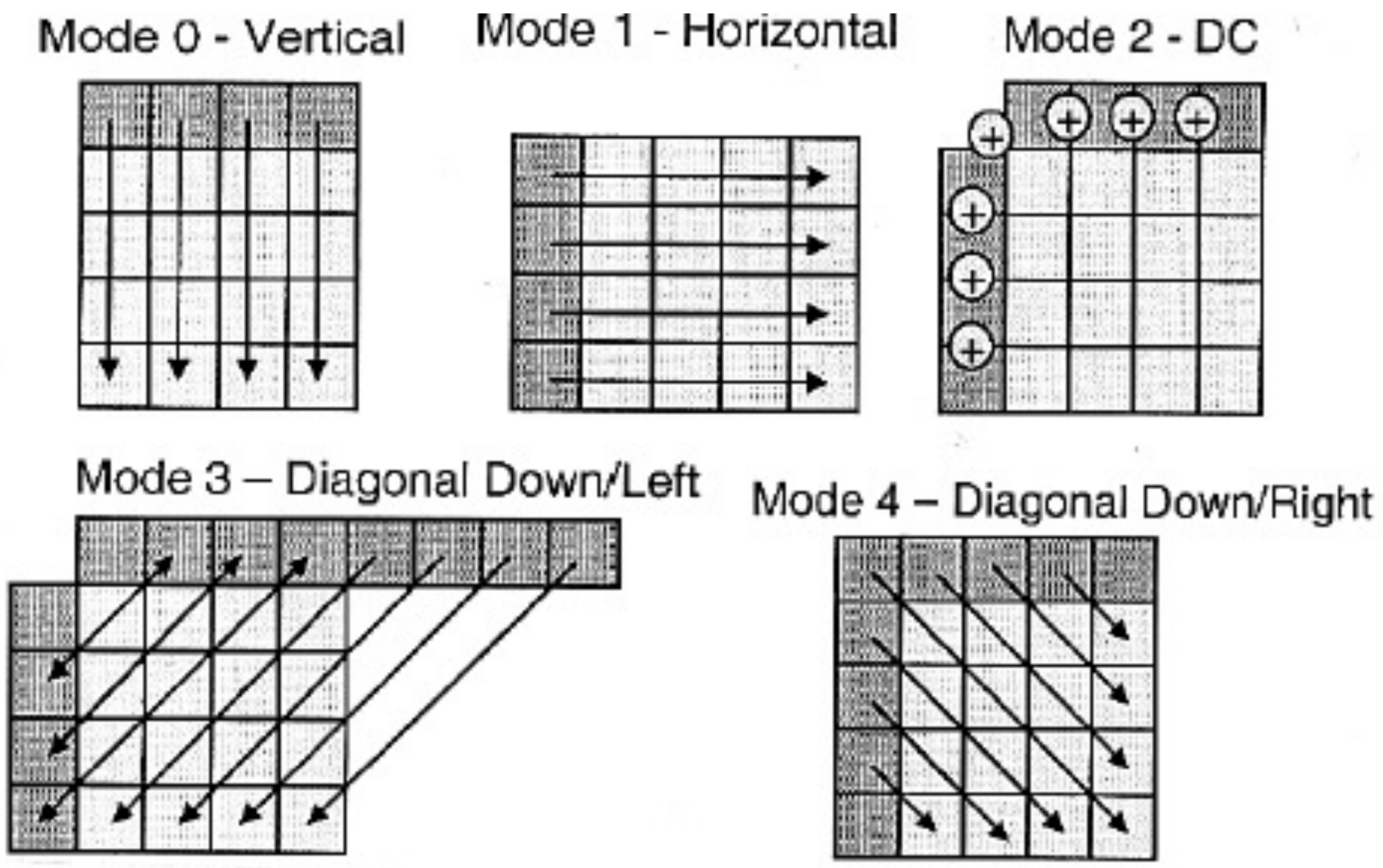


Fig. 11. Five of the nine Intra_4×4 prediction modes.

Outline

- Temporal prediction (inter-prediction)
  - Uni-directional, bi-directional prediction
- Spatial prediction (intra-prediction)
- ⇨ Transform coding of prediction error and coding of side information
- Rate-distortion optimization (RDO) for motion estimation and coding mode selection
- Rate control, Deblocking filtering
- GoP Structure
- Deep learning for image and video coding

Coding of Prediction Error Blocks

- Error blocks typically still have spatial correlation
- To exploit this correlation:
  - Vector quantization
  - Transform coding
- Vector quantization
  - Can effectively exploit the typical error patterns due to motion estimation error
  - Computationally expensive, requires training
- Transform coding
  - Can work with a larger block under the same complexity constraint
  - Which transform to use?
  - DCT is typically used

Transform Coding of Error Blocks

- Theory: Karhunen Loeve Transform is best possible block-based transform
- Problems with theory:
  - Finding an accurate model (covariance matrix) of the source is difficult
  - Model and KLT change over time and in different regions
  - Decoder and encoder need to use same KLT
  - Implementation complexity: a full matrix multiplication is necessary to implement KLT
- Practice: Discrete Cosine Transform
  - When the inter-pixel correlation approaches one, the KLT approaches the DCT

From Amy Reibman

Transform Coding: What block size?

- Theory: Larger transform blocks (using more pixels) are more efficient
- Problem with theory:
  - Hard to get an accurate model of the correlation of distant pixels
  - In the limit as the inter-pixel correlation approaches one, the KLT approaches the DCT; however, the inter-pixel correlation of distant pixels is not close to one
- Practice:
  - Small block transforms – usually 8x8 pixels, although in more recent systems 4x4 blocks and 16x16 blocks are also used, chosen adaptively!

From Amy Reibman

Key Idea in Video Compression

- Divide a frame into non-overlapping blocks
- Predict each block using different modes (intra-, unidirection-inter, bidirectional-inter)
  - Intra: choose best intra-prediction mode
  - Uni-Inter: choose best matching block (determine motion vector)
  - Bidirectional-inter: determine two motion vectors and matching blocks
- Choose the best prediction mode (mode-selection)
  - The one leading to least prediction error or best RD tradeoff
- Quantize and code prediction error using the transform coding method
  - Prediction errors have smaller variance than the original pixel values and can be coded with fewer bits
- Code (losslessly) the mode and motion info
  - Additional bits over coding the error
- Work on each block independently and can be speed up using parallel computation
- Hybrid coding: predictive coding+transform coding

Coding of Motion Vectors

- Typically we predict the MV for a current block from MV of previously coded blocks and code the prediction error using entropy coding
  - Ex: using median MV of the top left, top, and left blocks
  - Prediction error is typically small, and has a Laplacian distribution (reduced entropy than original MV!)
- We may use a special flag to indicate the case when MV=0 (a special mode)
- We may also use a special flag when MV=0, and the prediction error is all quantized to zero (Skip mode)

Outline

- Temporal prediction (inter-prediction)
  - Uni-directional, bi-directional prediction
- Spatial prediction (intra-prediction)
- Transform coding of prediction error and coding of side information
- ⇨ Rate-distortion optimization (RDO) for motion estimation and coding mode selection
- Rate control, Deblocking filtering
- GoP Structure
- Deep learning for image and video coding

Rate-distortion Optimized Motion Estimation

- In EBMA: we find MV to minimize the prediction error

$$\mathbf{v}^* = \arg\min\{D_p(\mathbf{v})\}, \quad D_p(\mathbf{v}) = \sum_{(x,y)\in B} \left|\psi_2(x,y) - \psi_1(x+v_x, y+v_y)\right|^p$$

- Why do we want to minimize prediction error?
  - # required bits for coding the error is proportional to mean square of prediction error
  - $$D_q = \varepsilon^2 \sigma_p^2 2^{-2R_p} \text{ (with MSE optimized quantizer design)}$$

    or more generally

    $$D_q = \varepsilon^2 \sigma_p^2 2^{-\alpha R_p} \text{ or } R_p = \frac{1}{\alpha} \log_2 \frac{\varepsilon^2 \sigma_p^2}{D_q}$$

- But we also need to code the MV!
  - A small error may be associated with a MV that rarely occur and hence require more bits to code!
- RD optimized motion estimation

$$\mathbf{v}^* = \arg\min\{D_p(\mathbf{v}) + \lambda R(\mathbf{v})\}$$

Variable Block Size Motion Estimation and Mode Decision

- For improved accuracy, starting with a maximum block size, we may partition it to smaller blocks, and allow different prediction modes (or MV / intra direction) in each sub-block.
- We pick the partition and mode for each subblock that yields minimal Lagrange cost for the entire block.

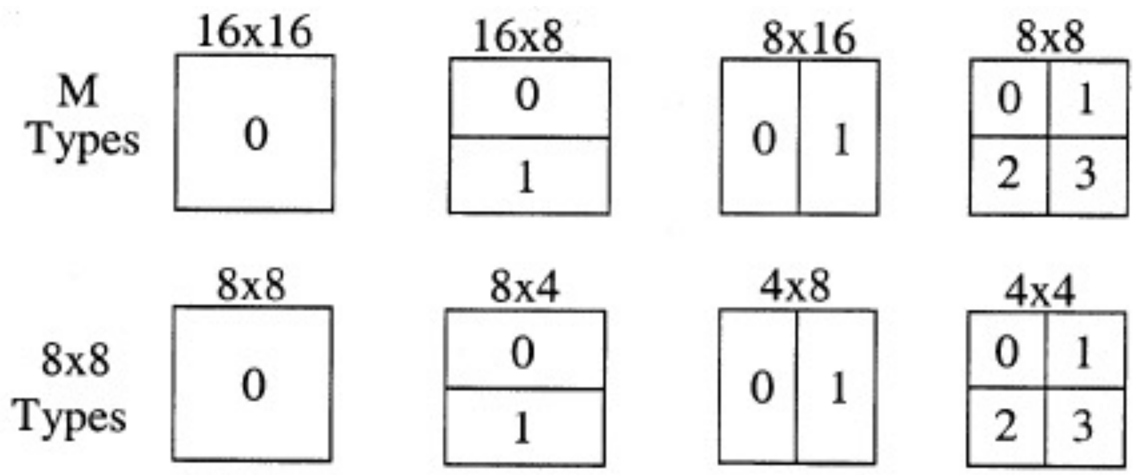


Fig. 12. Segmentations of the macroblock for motion compensation. Top: segmentation of macroblocks, bottom: segmentation of 8×8 partitions.

From [Wiegand2003]

Coding Mode Selection

- Which mode to use for a block?
  - Use RD optimal motion estimation to determine the block size and best MV for both unidirectional and bi-directional inter mode,
  - Use RD optimal intra-mode decision to determine best intra-mode
  - Then choose between P-mode, B-mode and I-mode
- Rate-distortion optimized (RDO) mode selection

$$m^* = \arg\min\{D_q(m) + \lambda R(m)\}$$

λ: Lagrangian multiplier, depending on expected quantization distortion or QP

$D_q(m)$: final reconstruction error with mode m =quantization error of prediction error

$R(m)$: total bits for mode m, including bits for coding the mode info, MV, and prediction error

RDO mode selection: Coding a block with all candidates modes and taking the mode that yields the least cost. Computationally expansive ☹

Fast mode selection: Using some simple computation to limit the candidate set, estimate the bits instead of running actual encoder,…

Pop Quiz

- Why use spatial prediction?
- How to perform spatial prediction?
- When will spatial prediction be better than temporal prediction?
- How to select which prediction mode to use?
- What do we need to specify in the bit stream?
- How do we code the prediction error?

Pop Quiz (w/ Answers)

- Why use spatial prediction?
  - As with temporal prediction: to reduce the variance of the signal to be coded
- How to perform spatial prediction?
  - Simple linear prediction
  - Adaptive directional prediction
- When will spatial prediction be better than temporal prediction?
  - Uncovered regions or when temporal prediction is less accurate
  - Complete scene change
- How to select which prediction mode to use?
  - Whichever leads to lower rate-distortion under a chosen λ
  - Under same distortion, whichever leads to lower bit rate, or vice versa
- What do we need to specify in the bit stream?
  - Side information: Coding mode and associated parameters (direction for Intra, MV for P or B). Need to be coded losslessly.
  - Prediction error (coded with quantization error)
- How do we code the prediction error?
  - Transform coding: transform the error block, quantize coefficients, entropy coding

Outline

- Temporal prediction (inter-prediction)
  - Uni-directional, bi-directional prediction
- Spatial prediction (intra-prediction)
- Transform coding of prediction error and coding of side information
- Rate-distortion optimization (RDO) for motion estimation and coding mode selection
- ⇨ Rate control and Deblocking filtering
- GoP Structure
- Deep learning for image and video coding

Rate Control: Why

- The coding method necessarily yields variable bit rate
- Active areas (with complex motion and/or complex texture) are hard to predict and requires more bits under the same QP
- Rate control is necessary when the video is to be sent over a constant bit rate (CBR) channel, where the rate when averaged over a short period should be constant
- Rate control is also necessary when the channel bandwidth is dynamically changing. Should code the video to a rate below the sustainable throughput
- Coded bitstream is smoothed by a buffer at the encoder output
  – Encoded bits (variable rates) are put into a buffer, and then drained at a constant rate or sustainable rate
  – The encoder parameter (QP, frame rate) need to be adjusted so that the buffer does not overflow or underflow

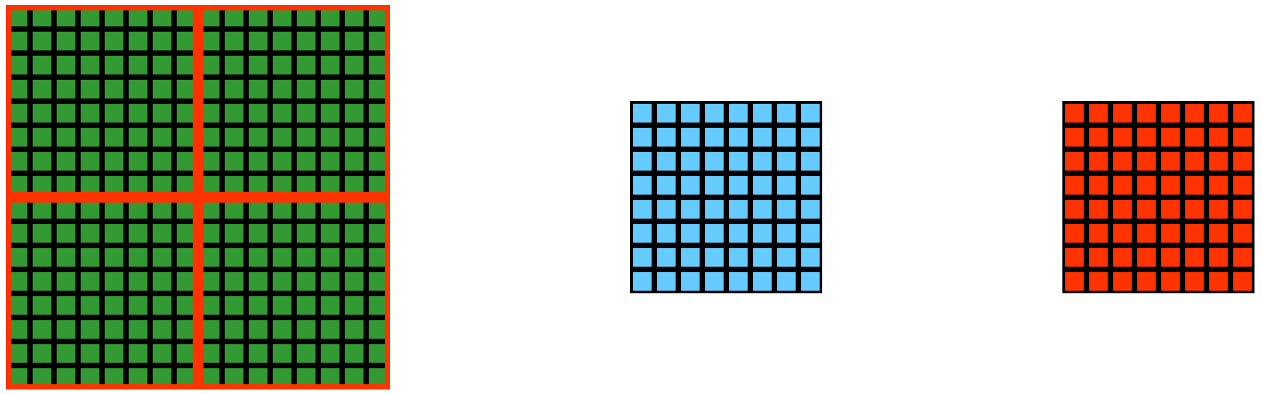
Rate Control: How

- General ideas:
  - Step 1) Determine the target rate at the frame level, based on the current buffer fullness
  - Step 2) Satisfy the target rate by varying QP
    - Average QP is determined at the frame level to meet the bit budget for this frame
    - QP is further adapted at block level based on remaining bits in this frame
    - Determination of QP requires an accurate model relating rate with q (quantization stepsize) (QP has a fixed relation with q)
    - Model used in MPEG2: $R \sim A/q + B/q^2$
  - A frame may be skipped if when the buffer is nearly full
- A complex problem in practice

In-Loop Filtering (Deblocking)

- Errors in previously reconstructed frames (mainly blocking artifacts) accumulate over time with motion compensated temporal prediction
  - Reduce prediction accuracy
  - Increase bit rate for coding new frames
- In-Loop filtering:
  - Filter the reference frame before using it for prediction
  - Must be done in the same way both at the encoder and decoder (in-loop, not postprocessing outside the encoder)
  - Can be embedded in the motion compensation loop
    - Half-pel motion compensation
    - Overlapped block motion compensation (OBMC)
  - Explicit deblocking filtering: removing blocking artifacts after decoding each frame
- In-Loop filtering can significantly improve coding efficiency
- Simple fixed filters lead to blurring!
- Complex adaptive deblocking filtering is used in H.264/HEVC

Macroblock Structure in 4:2:0 Color Format



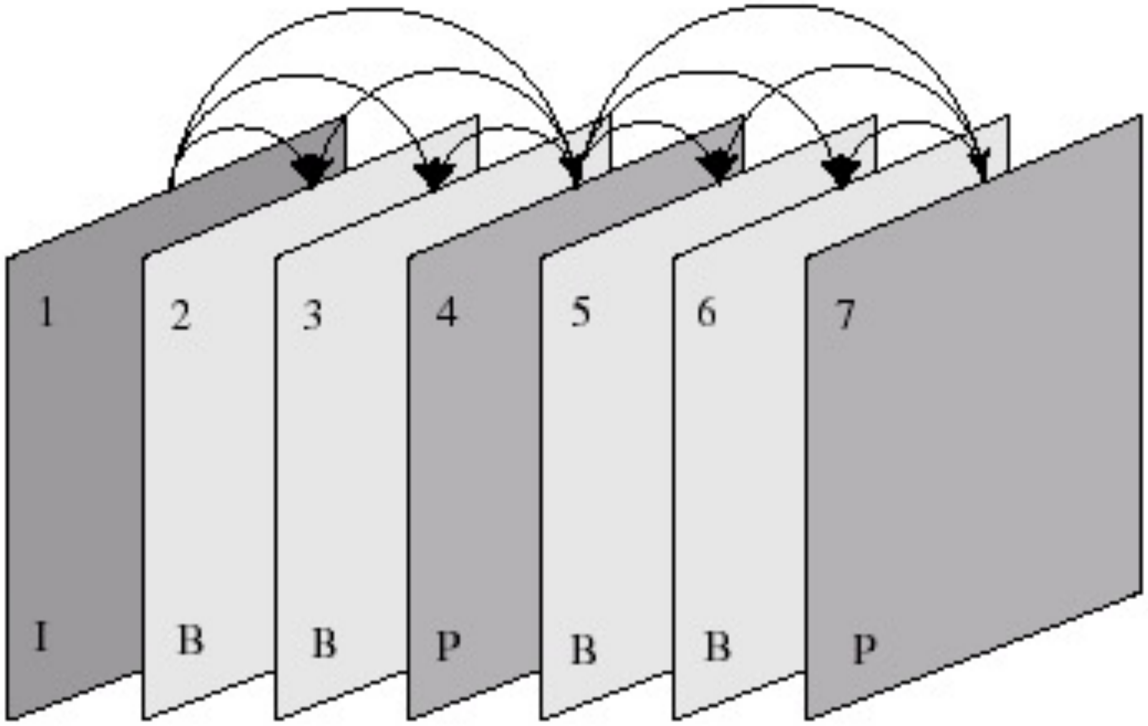
4 8x8 Y blocks   1 8x8 Cb blocks   1 8x8 Cr blocks

Typically we use the 16x16 Y blocks to determine the best MV and intra-prediction direction and apply the resulting MV/intra-direction to both Y and Cb and Cr.

Outline

- Temporal prediction (inter-prediction)
  - Uni-directional, bi-directional prediction
- Spatial prediction (intra-prediction)
- Transform coding of prediction error and coding of side information
- Rate-distortion optimization (RDO) for motion estimation and coding mode selection
- Rate control, Deblocking filtering
- GoP Structure
- Deep learning for image and video coding

Group of Picture Structure

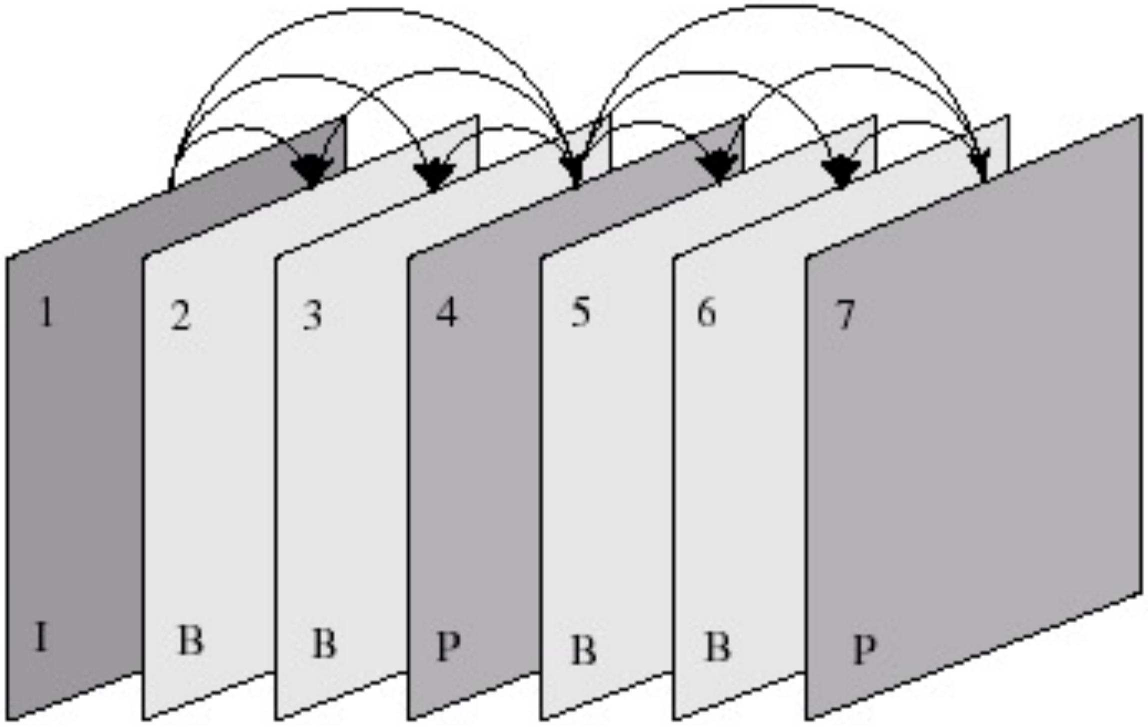


Encoding order: 1 4 2 3 7 5 6

Group-of-picture (GoP) structure

- I-frames coded without reference to other frames
  - Only intra prediction allowed
  - To enable random access (AKA channel change), fast forward, stopping error propagation
- P-frames coded with reference to previous frames only
  - Uni-directional inter-prediction or intra mode
  - Requires more computation
  - Can cause transmission error propagation
- B-frames coded with reference to previous and future frames
  - Bi-directional or uni-directional inter-prediction or intra mode
  - Highest coding efficiency
  - Requires more computation and extra delay!
  - Enable frame skip at receiver (temporal scalability)
- Typically, an I-frame every 1-2 sec.
- Typically, two B frames between each P frame
  - Compromise between compression and delay

Delay due to B-Frames



Encoding order: 1 4 2 3 7 5 6

Encoding delay=time when a frame is encoded--time when a frame arrives
First B-frame: 2 * frame interval + encoding time for 1 P and 1 B
Second B-frame: 1 * frame interval + encoding time for 1 P and 2 B
B-frame is usually not used in real-time applications (video phone/conferencing, gaming, virtual desktop, etc.)

Pseudo Code for Coding an I-frame

```
%Assume: f: current frame to be coded; N: block size for prediction, q: quantization stepsize
Function [fQ]=IframeCoding(f,q)
for (x0=1:N:height, y0=1:N:width) %for every NxN block B at (x0,y0) in f2
    B=f(x0:x0+N-1,y0:y0+N-1);
    [BI, intramode,errI]=intraPred(B,f,x0,y0);%Find best intra-prediction for B, based on previously coded
    pixels in this frame, BI is the predicted block, intramode is the chosen intra mode, errI is the prediction
    error (e.g. sum of absolute difference)
    BP=BI;
    Bits=BinaryCodingModeIFrame(intramode); %entropy coding to generate binary bits for representing the
    chosen mode and associated side information: intramode
    AppendBits(Bits); %append these bits to an existing bit stream
    BE=B-BP; %form prediction error block to be coded using the best prediction
    %transform coding of BE, assume transform is to be done at smaller block size (e.g. 8x8) than prediction
    block (e.g. 16x16)
    For every subblock BE(i) in BE
        T=dct2(BE(i))
        TQI=quantize(T,q); %generating quantizer indices using quantization stepsize q, e.g.
                                        %TQI=floor(T+q/2)/q
        Bits=BinaryCodingCoef(TQI); %entropy coding to generate binary bits for quantization indices
        AppendBits(Bits);
        TQ=dequantize(TQI,q); %generate quantized values from quantizer indices, e.g. TQ=TQI*q;
        BEQ(i)=idct2(TQ); %reconstruct the subblock from quantized DCT coefficients
    BEQ=Assemble(BEQ(1),BEQ(2),…) %put all quantized subblocks to a larger block
    fQ(x0:x0+N-1,y0:y0+N-1)=BP+BEQ;
end; end;
```

Pseudo Code for Coding a P-frame

```
Assume: f1: previous frame (already coded and decoded); f2: current frame
Function [f2Q]=PframeCoding(f2,f1,q)
for (x0=1:N:height, y0=1:N:width) %for every NxN block B at (x0,y0) in f2
        B=f2(x0:x0+N-1,y0:y0+N-1);
        [BI, intramode,errI]=intraPred(B,f2,x0,y0);%Find best intra-prediction for B, based on previously coded
        pixels in this frame
        [BP1,MV,errP]=EBMA(B,x0,y0,f1); %Find best prediction block BP1 for B in f1, and corresponding MV
        “MV” and matching error “errP”
        [BP,mode]=modeDecisionPframe(errI,errP,intramode, MV,B,BI,BP1)
        %Choose mode based on errI,errP as well as MV,intramode,
        %In the simplest case where rates are not considered, simply compare errI and errP and choose the one
        with smaller error. If (errI<=errP) mode=“intra”, BP=BI, else mode=“interP”, BP=BP1;
        % More generally, the coder may find the bits needed and the corresponding quantization error to code in
        the intra mode, and in the inter mode and choose the one with the smallest Lagrangian cost

        Bits=BinaryCodingModePFrame(mode,intramode, MV); %entropy coding to generate binary bits for
        representing the chosen mode and associated side information: intramode or MV
        AppendBits(Bits); %append these bits to existing bit stream
        BE=B-BP; %form prediction error block to be coded using the best prediction

        %Transform coding of BE, Same as in IframeCoding( )
        …..
         f2Q(x0:x0+N-1,y0:y0+N-1)=BP+BEQ;
end; end;
```

Pseudo Code for Coding a B-frame

Assume: f1: a previous frame (already coded and decoded); f2: current frame; f3: a future frame (already coded and decoded), N: block size for prediction

```
Function [f2Q]=BframeCoding(f2,f1,f3,q)
for (x0=1:N:height, y0=1:N:width) %for every NxN block B at (x0,y0) in f2
          B=f2(x0:x0+N-1,y0:y0+N-1);
          [BI, intramode,errI]=intraPred(B,f2,x0,y0);%Find best intra-prediction for B, based on previously coded
          pixels in this frame
          [BP1,MV1,errP1]=EBMA(B,x0,y0,f1); %Find best prediction block BP1 for B in f1, and corresponding MV
          “MV1”
          [BP2,MV2,errP2]=EBMA(B,x0,y0,f3);%Find best prediction block BP2 for B in f2, and corresponding MV
          “MV2”
          BB=(BP1+BP2)/2; % for bi-direcitional prediction.
          errB=sum(sum(abs(B-BB)));
          [BP,mode]=modeDecisionBframe(errI,errP1,errB,intramode, MV1,MV2,B,BI,BP1,BB)
          %Choose mode based on errl,errP1,errB as well as MV1,MV2,intramode, BP is the chosen prediction.
          %For example, it can choose the one giving the least prediction error
          Bits=BinaryCodingModeBFrame(mode,intramode, MV1,MV2); %entropy coding to generate binary bits for
          representing the chosen mode and associated side information: intramode or MV1 and/or MV2
          AppendBits(Bits); %append these bits to existing bit stream
          BE=B-BP; %form prediction error block to be coded using the best prediction
          %Same as IframeCoding( ) for transform, quantized and code BE to generate BEQ
          …..
          f2Q(x0:x0+N-1,y0:y0+N-1)=BP+BEQ;
end; end;
```

Pseudo Code for Coding a GoP

Consider coding the following frames f1,f2,f3,f4,f5,f6,f7,f8,f9,f10, in IBBPBBPBBI … structure

Note: encoding order: f1 (I), f4(P), f2(B), f3(B), f7(P), f5(B), f6(B), …

```
[f1Q]=IframeCoding(f1,q);
[f4Q]=PframeCoding(f4,f1Q,q); %note: use f1Q for prediction!
[f2Q]=BframeCoding(f2,f1Q,f4Q,q); %note: Use f1Q and f4Q for prediction!
[f3Q]=BframeCoding(f3,f1Q,f4Q,q);

[f7Q]=PframeCoding(f7,f4Q,q);
[f5Q]=BframeCoding(f5,f4Q,f7Q,q);
[f6Q]=BframeCoding(f6,f4Q,f7Q,q);

[f10Q]=IframeCoding(f10,q);
[f8Q]=BframeCoding(f8,f7Q,f10Q,q);
[f9Q]=BframeCoding(f9,f7Q,f10Q,q)
…
```

Pop Quiz

- What is the problem if we only code the first frame as I-frame and all remaining frames as P-frame?
- What are the benefits of dividing frames into GOPs and use different coding modes for different frames in a GOP?
- What are the potential problems with using the GOP structure?

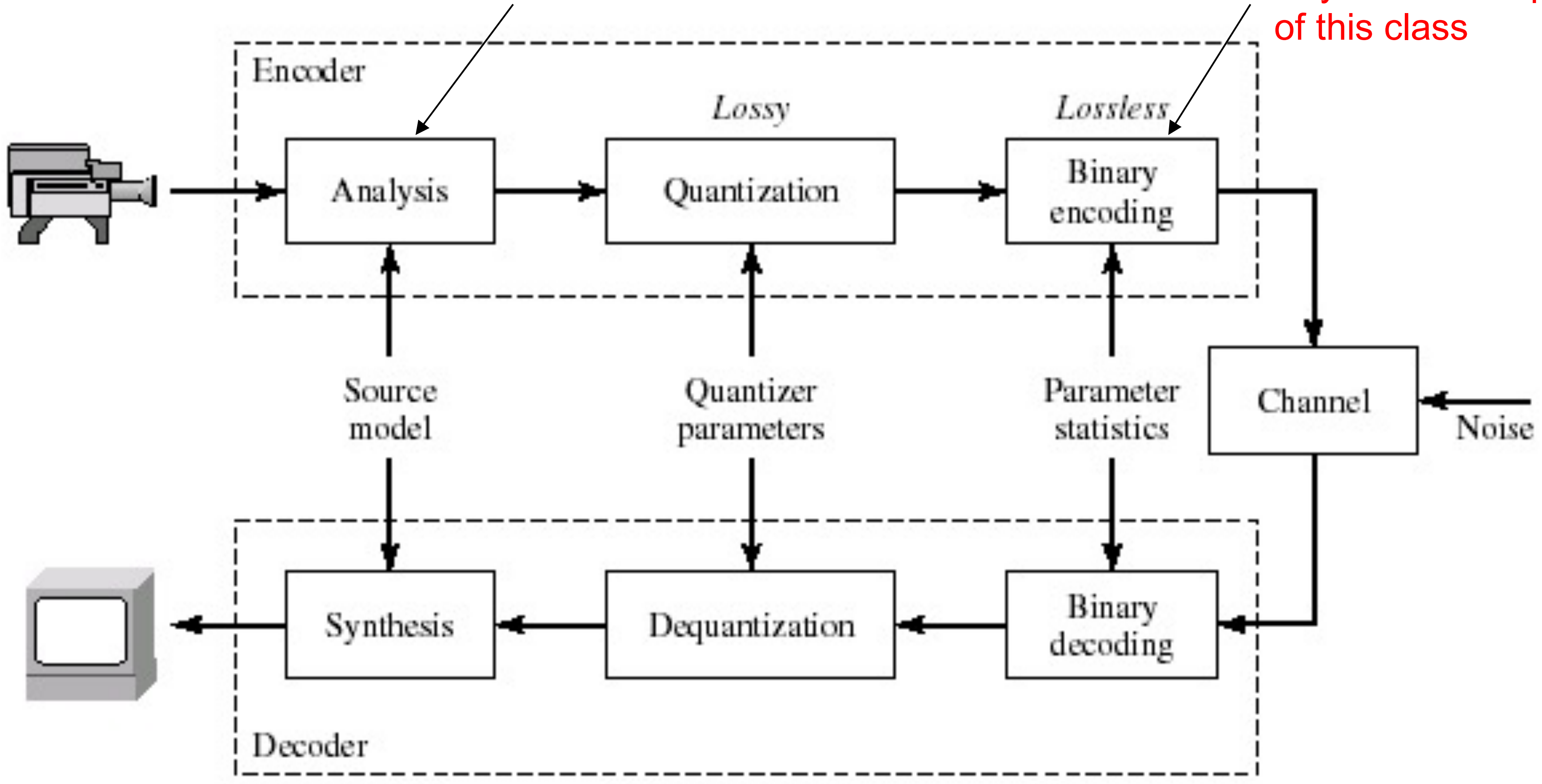
Pop Quiz (w/ Answers)

- What is the problem if we only code the first frame as I-frame and all remaining frames as P-frame?
  - Does not facilitate channel surfing
  - Does not facilitate fast playback
  - If there are some bit errors during transmission in one frame, all future frames will be decoded wrong (transmission error propagation!) ☹
- What are the benefits of dividing frames into GOPs and use different coding modes for different frames in a GOP?
  - Allow channel surfing
  - Allow fast playback with different playback speed
  - I frame stops error propagation
  - B frame improves coding efficiency
- What are the potential problems with using the GOP structure?
  - Encoding delay, decoding delay
  - Bursty bit stream (periodic rate spikes, which will lead to variable transmission delay)
  - Effective for on-demand streaming or live streaming, but not for real-time interactive applications

Summary: Components in a Coding System

Prediction, transform

Very important!
Beyond the scope
of this class



Outline

- Temporal prediction (inter-prediction)
  - Uni-directional, bi-directional prediction
- Spatial prediction (intra-prediction)
- Transform coding of prediction error and coding of side information
- Rate-distortion optimization (RDO) for motion estimation and coding mode selection
- Rate control, Deblocking filtering
- GoP Structure
- ⇨ Deep learning for image and video coding

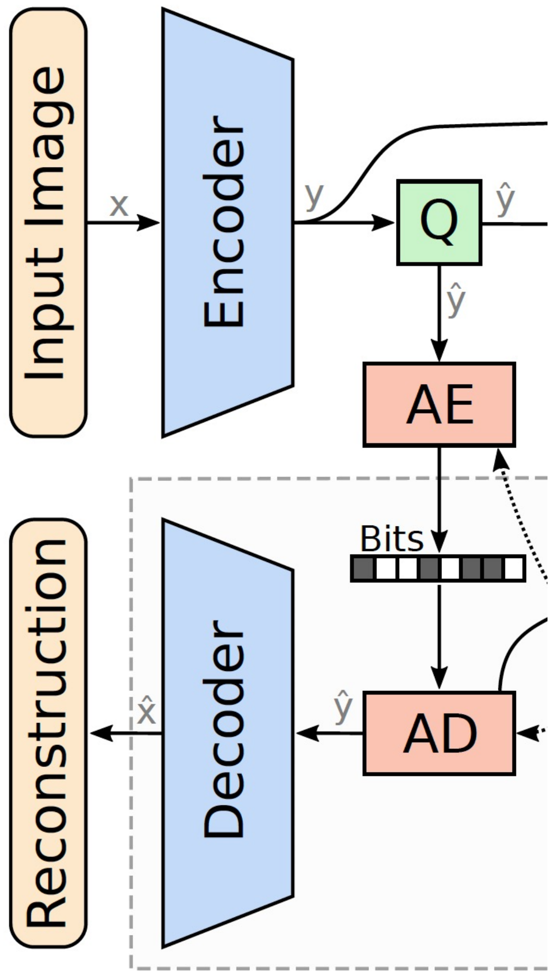
Deep learning for Image and Video Compression

- Using deep learning for certain modules
  - Intra-prediction
  - Inter-prediction
  - Deblocking filtering
  - Upsampling for fractional pel motion compensation
  - …
- End-to-end leant image compression
  - Learn and code image features!
- End-to-end leant video compression
  - Learn and code motion and residual image features!

Learnt Image Compression (not required)

- Key ideas:
  - Learn compact features which can be used to reconstruct an image and can be coded efficiently
    - Auto encoder structure: Encoder generates the latent features from an image; Decoder reconstructs the image from the features
  - Use weighted average of distortion and rate as loss function
  - Use entropy of the quantized latent variables to approximate the rate.
    - Entropy depends on the pdf of the quantized latents
  - Quantization is not differentiable. Use differentiable proxy during training
    - Eg. Adding uniform noise.
    - Pdf of quantized variable = convolution of the pdf of the original latent variable and uniform distribution

First Generation: no probability distribution prediction



y: latent features
$\hat{y}$: quantized latents, quantization by rounding

During training:
assume $\hat{y}$=y+q where q is a uniform noise in (-½, ½)
For entropy calculation:
Pdf of $\hat{y}$ is the convolution of y and uniform pdf.

Training loss:

$$L = \underbrace{\lambda E\{\|x - \hat{x}\|_2^2\}}_{\text{Distortion}} - \underbrace{E\{\log p(\hat{y})\}}_{\text{Rate estimation by entropy}}$$

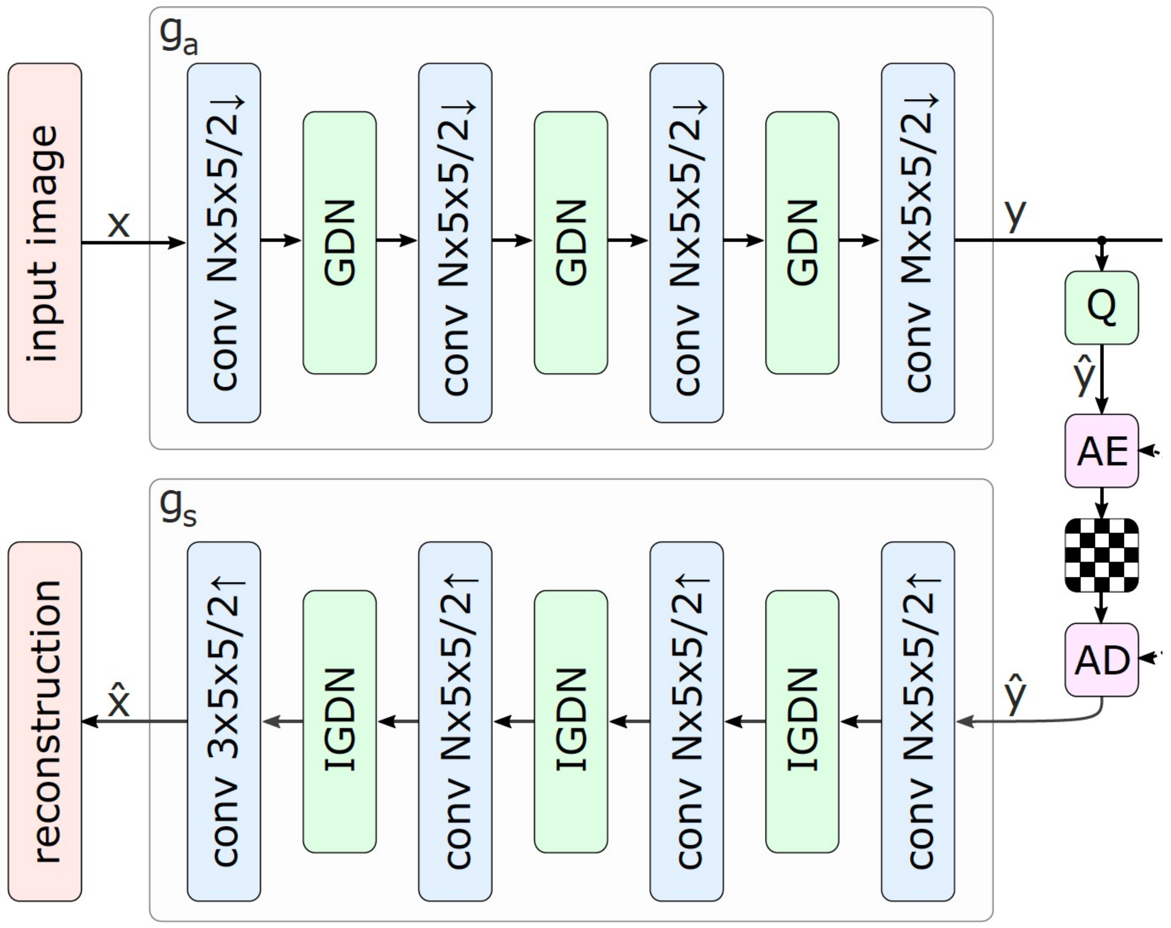
Can use other distortion or quality metrics

λ controls the tradeoff between rate and distortion
Train different networks for different λ
-> different models for different rates

Significant improvement over JPEG and JPEG2000, but also significant computation overhead

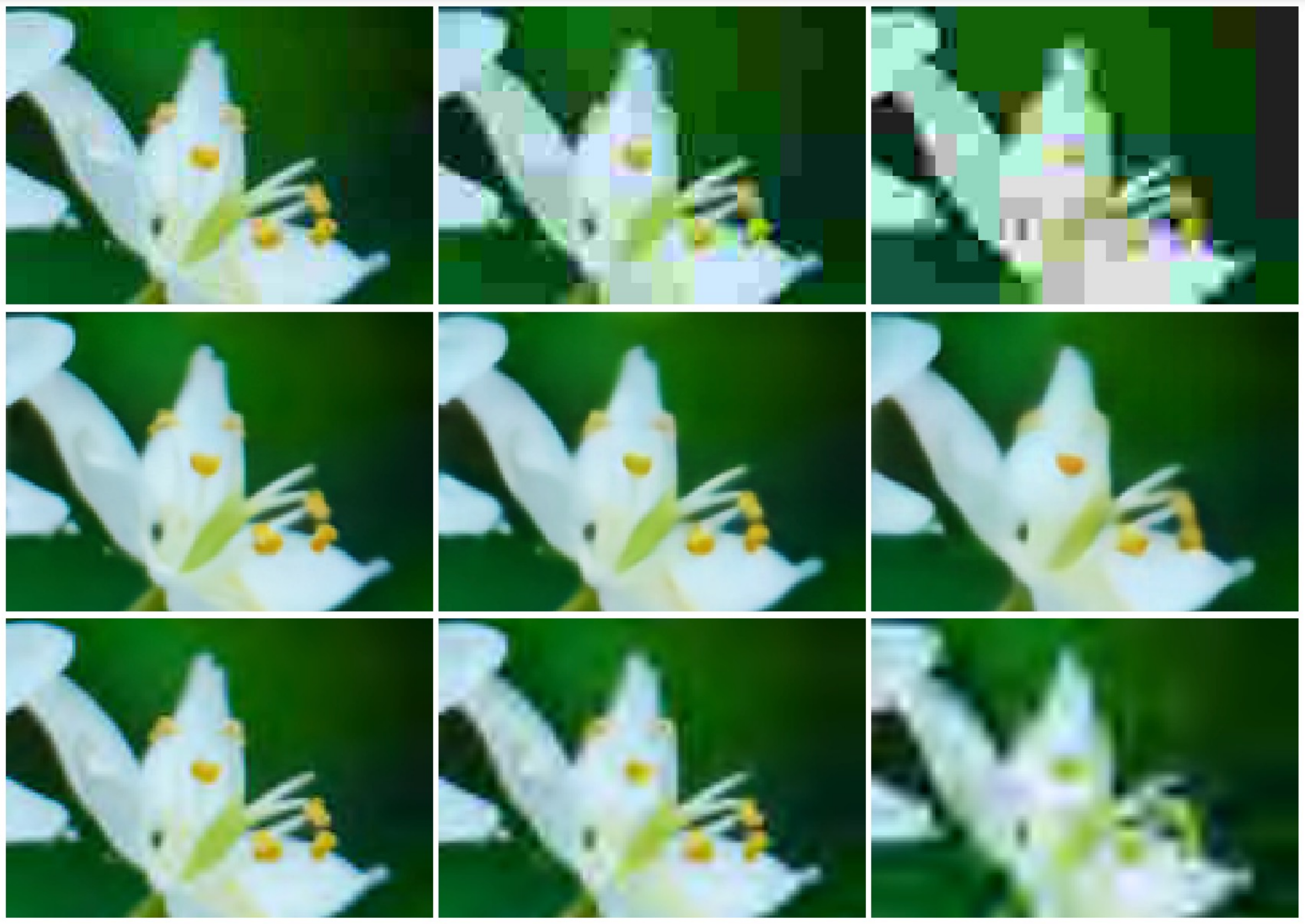
End-to-end Optimized Image Compression J Ballé, V Laparra, EP Simoncelli
International Conference on Learning Representations (ICLR) 2017. https://arxiv.org/pdf/1611.01704.pdf

Model by Balle et al 2017



End-to-end Optimized Image Compression J Ballé, V Laparra, EP Simoncelli
International Conference on Learning Representations (ICLR) 2017

Perceptual Quality Comparison



From End-to-end Optimized Image Compression J Ballé, V Laparra, EP Simoncelli
International Conference on Learning Representations (ICLR) 2017. https://arxiv.org/pdf/1611.01704.pdf

Objective Quality Comparison

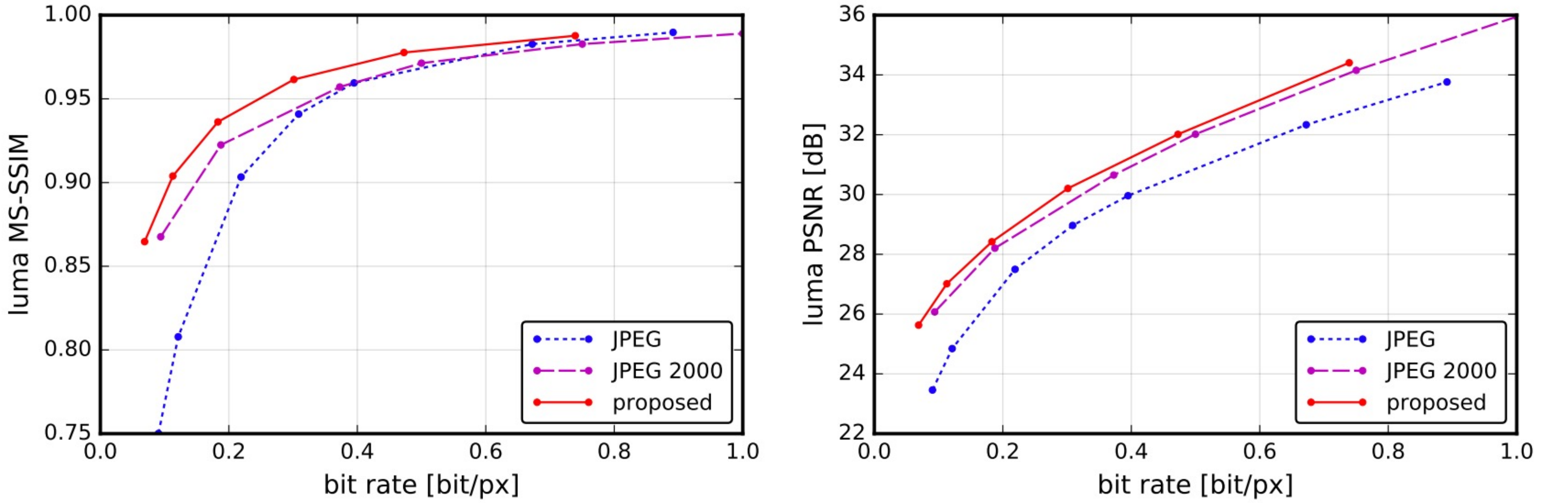


Figure 7: Rate–distortion curves for the luma component of image shown in figure 5. Left: perceptual quality, measured with multi-scale structural similarity (MS-SSIM; Wang, Simoncelli, and Bovik (2003)). Right: peak signal-to-noise ratio ($10\log_{10}(255^2/\text{MSE})$).

From: End-to-end Optimized Image Compression J Ballé, V Laparra, EP Simoncelli
International Conference on Learning Representations (ICLR) 2017. https://arxiv.org/pdf/1611.01704.pdf

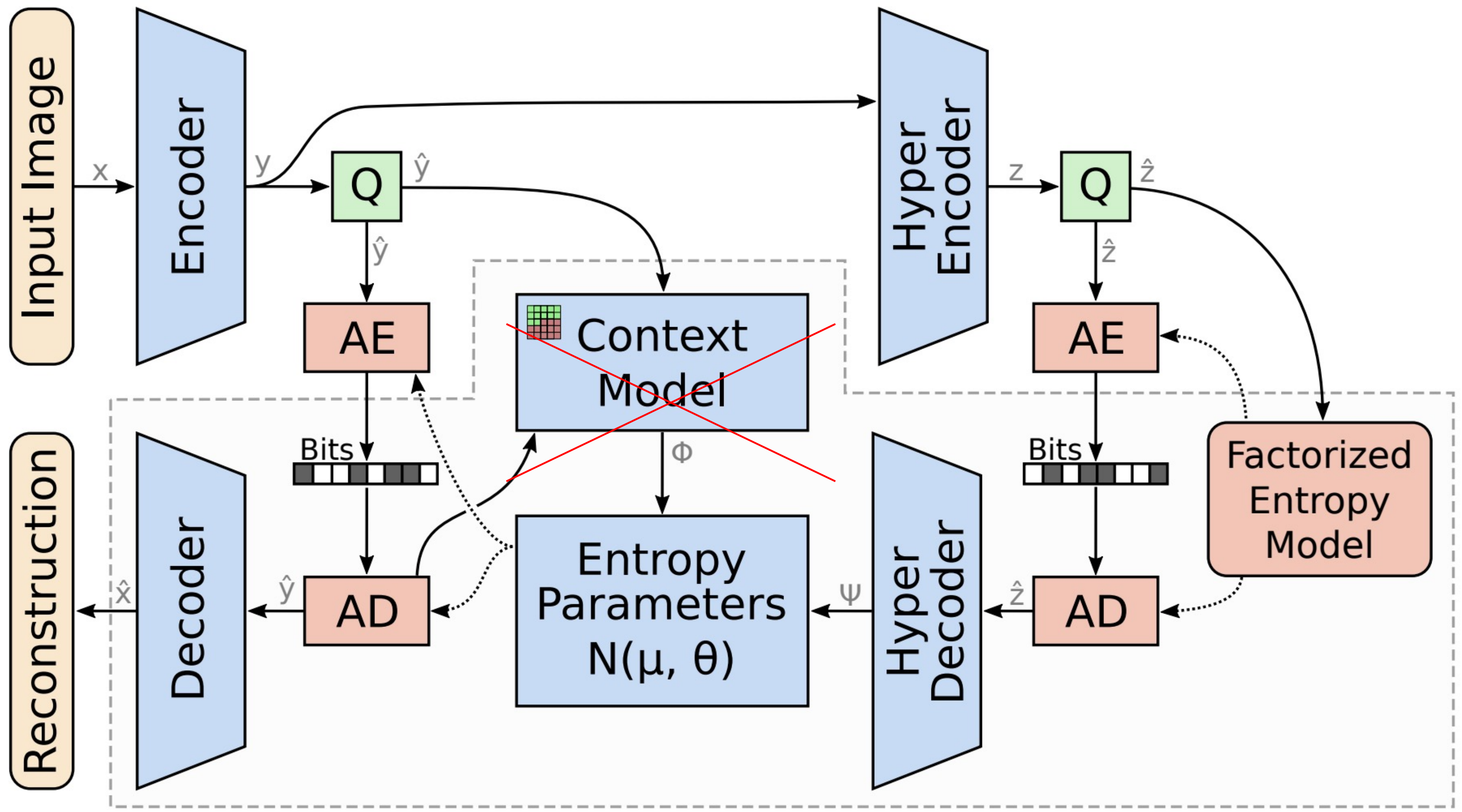
SSIM and MS-SSIM: objective metric that matches more closely with perceived quality than PSNR!

Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in Signals, Systems and Computers, 2004. Conference Record of the Thirty-Seventh Asilomar Conference on*, IEEE, vol. 2, 2003, pp. 1398–1402.*

Second and Third Generation: Predict the conditional distribution of the latents

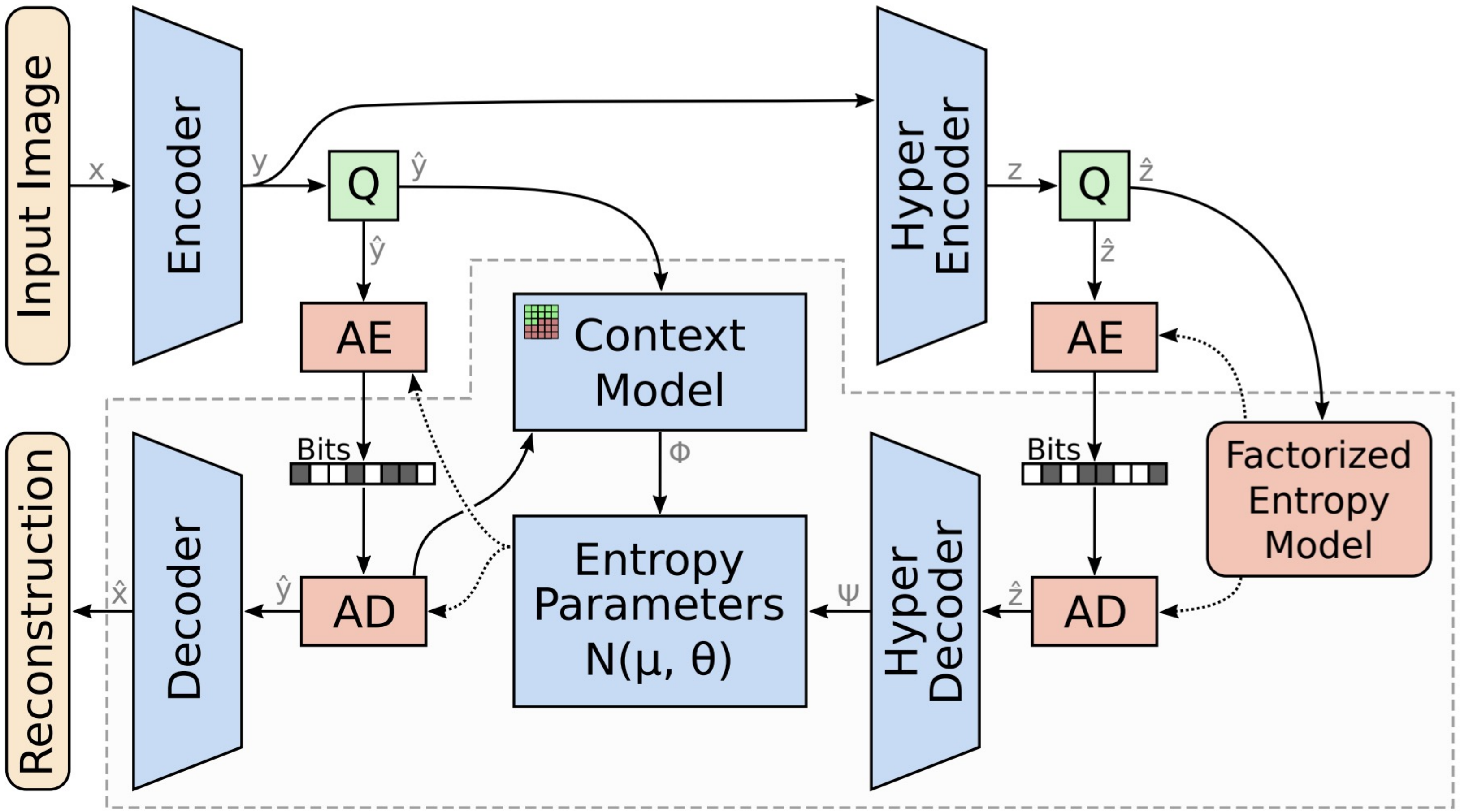
- Use Context-Based Arithmetic Coding (CABAC), which codes each latent variable based on the conditional probability of the variable given other variables already coded.
  - Conditional probability =1 -> requires no bits.
  - Goal: parameterize the conditional distribution (e.g. Gaussian), and estimate its parameters (e.g. mean and STD) so that the actual variable has high probability.
- First generation: no prediction of probability distribution
- Second generation: generate and code “hyper priors” to predict probability distribution parameters of latent features
- Third generation: using autoregressive context (previously coded features) and hyper priors to predict the probability distribution parameters

Second Generation



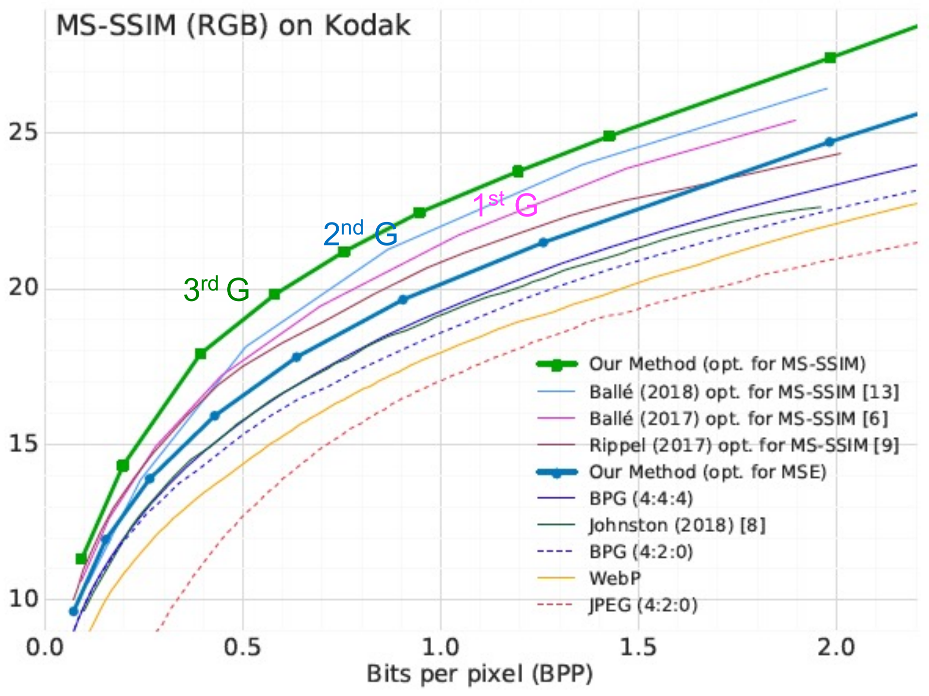
Variational image compression with a scale hyperprior J Ballé, D Minnen, S Singh, SJ Hwang, N Johnston International Conference on Learning Representations (ICLR) 2018. https://arxiv.org/pdf/1802.01436.pdf

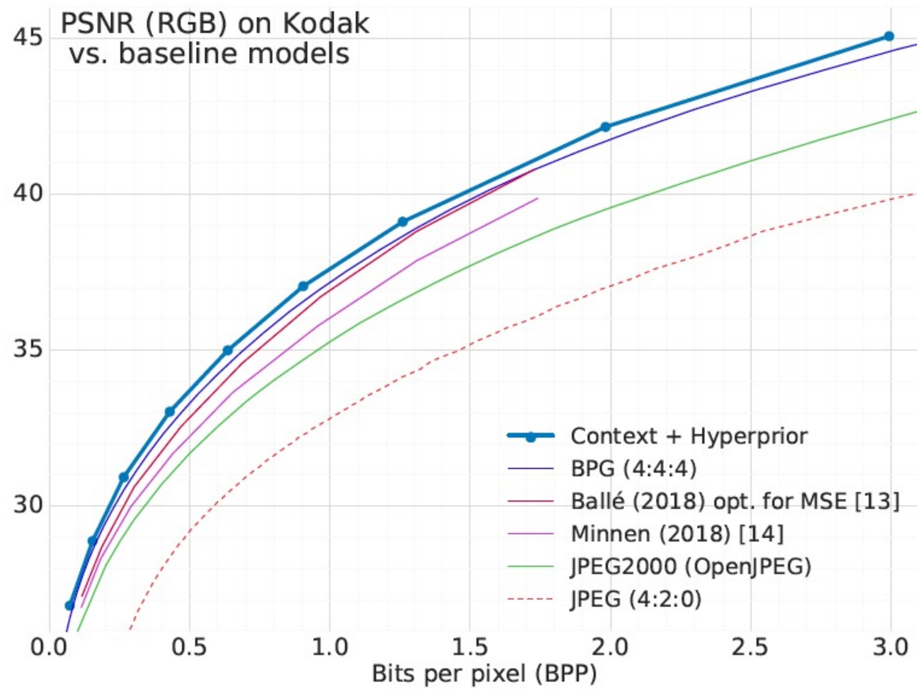
Third Generation



Joint autoregressive and hierarchical priors for learned image compression D Minnen, J Ballé, GD Toderici, Advances in Neural Information Processing Systems (NeurIPS) 31, 10771-10780. https://arxiv.org/pdf/1809.02736.pdf

Objective Quality Comparison

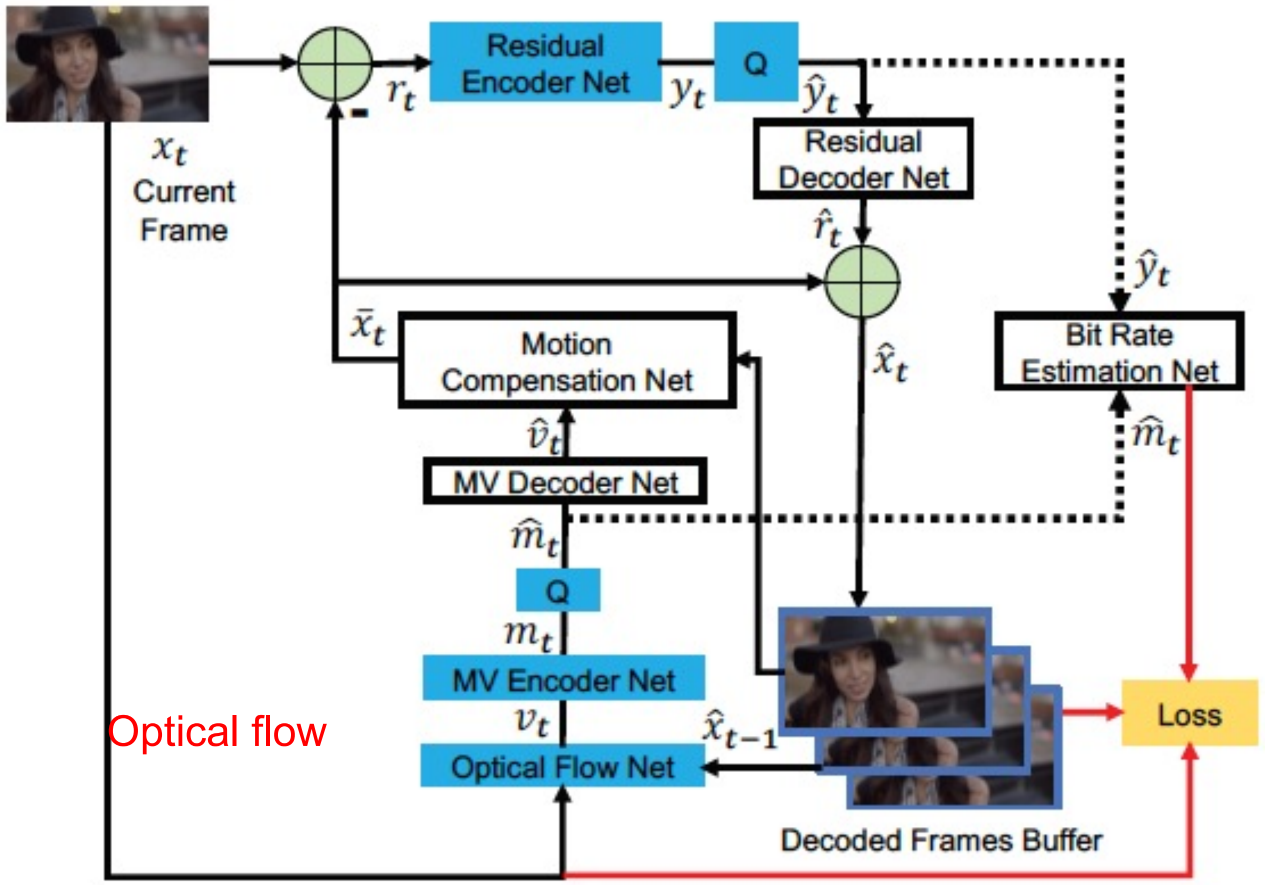




BPG: Intra coding method in HEVC video coder (Currently best standard image coder in terms of PSNR)

F. Bellard, BPG image format (http://bellard.org/bpg/), Accessed: 2017-01-30. [Online]. Available: http://bellard.org/bpg/.

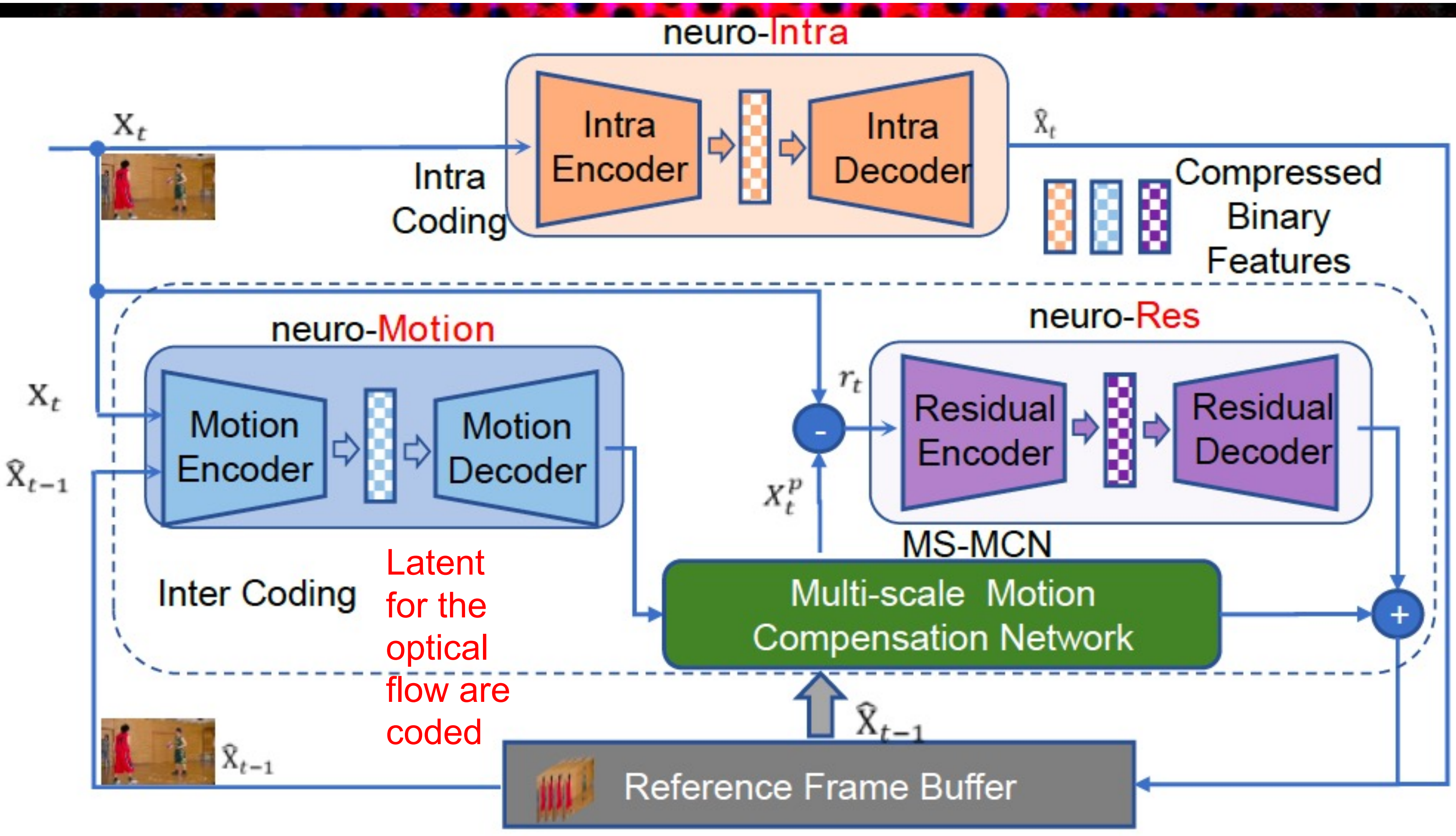
Learnt Video Coding: 1st G



Optical flow and residual image are coded using a learnt image coder

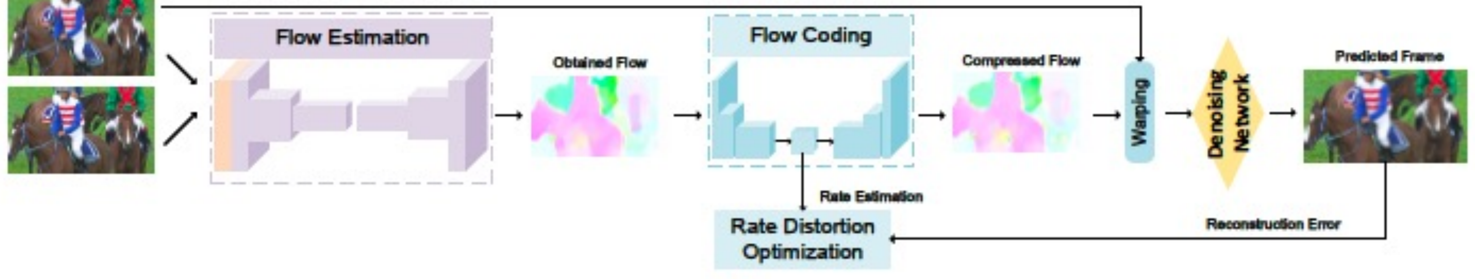
G. Lu, W. Ouyang, D. Xu, X. Zhang, C. Cai, and Z. Gao, "Dvc: An endto- end deep video compression framework," in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.

Learnt Video Coding: 2nd G



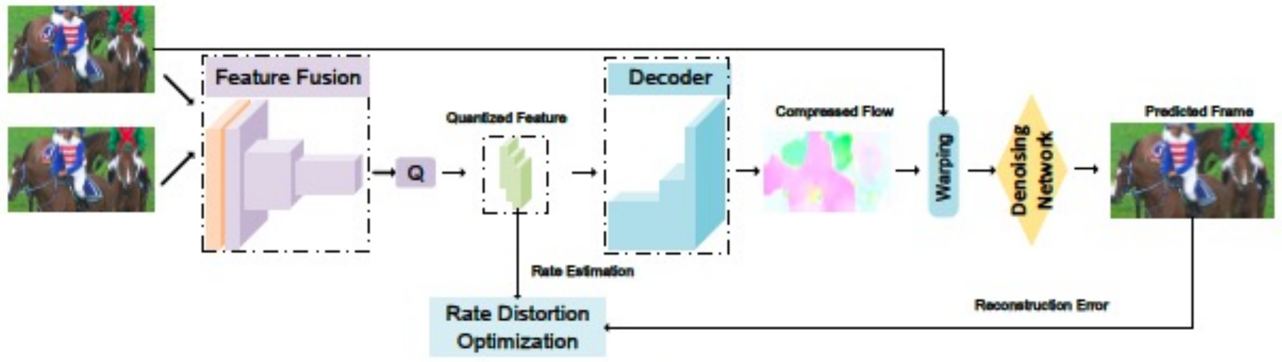
Latent for the optical flow are coded

From H. Liu *et al*., "Neural Video Coding using Multiscale Motion Compensation and Spatiotemporal Context Model," in *IEEE Transactions on Circuits and Systems for Video Technology*, doi: 10.1109/TCSVT.2020.3035680. https://ieeexplore-ieee-org.proxy.library.nyu.edu/stamp/stamp.jsp?tp=&arnumber=9247134



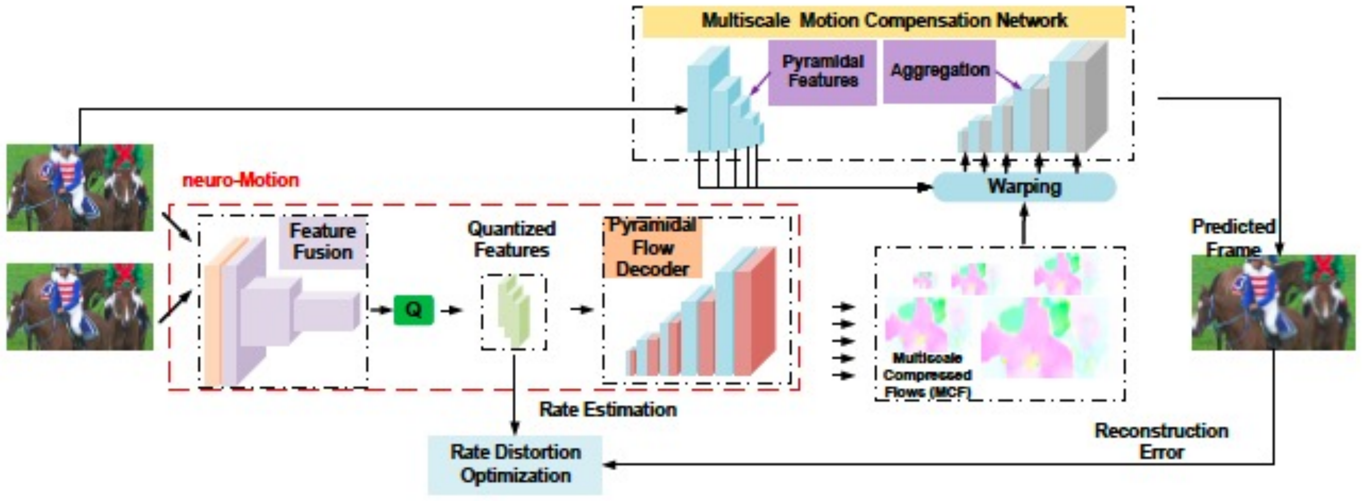
(a)

First estimate flow, then code



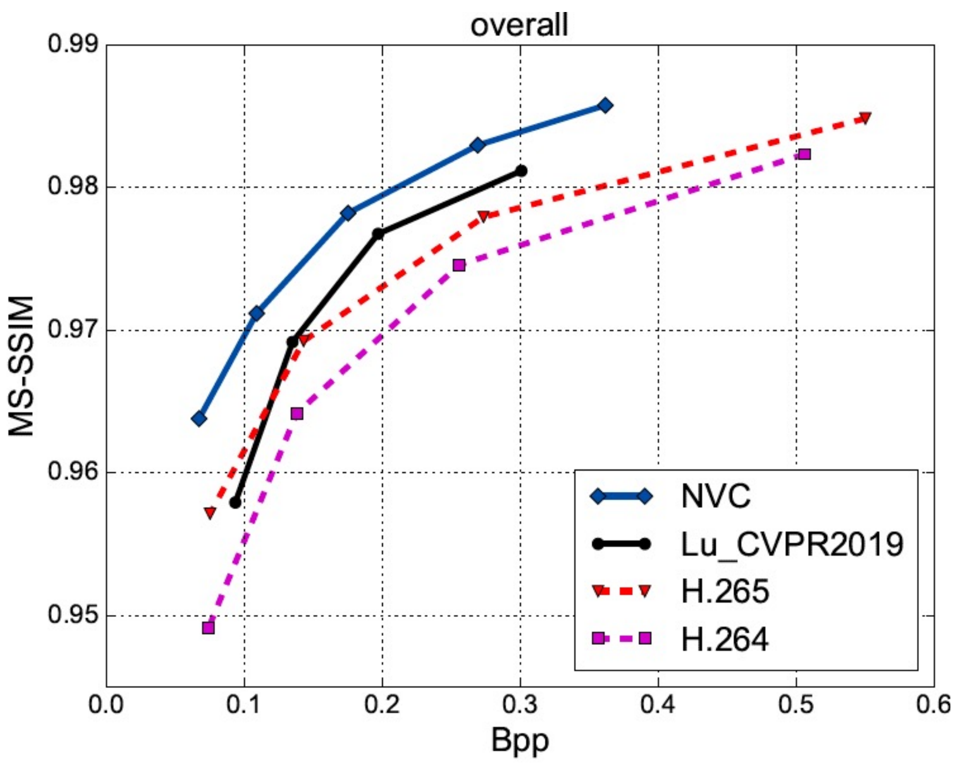
(b)

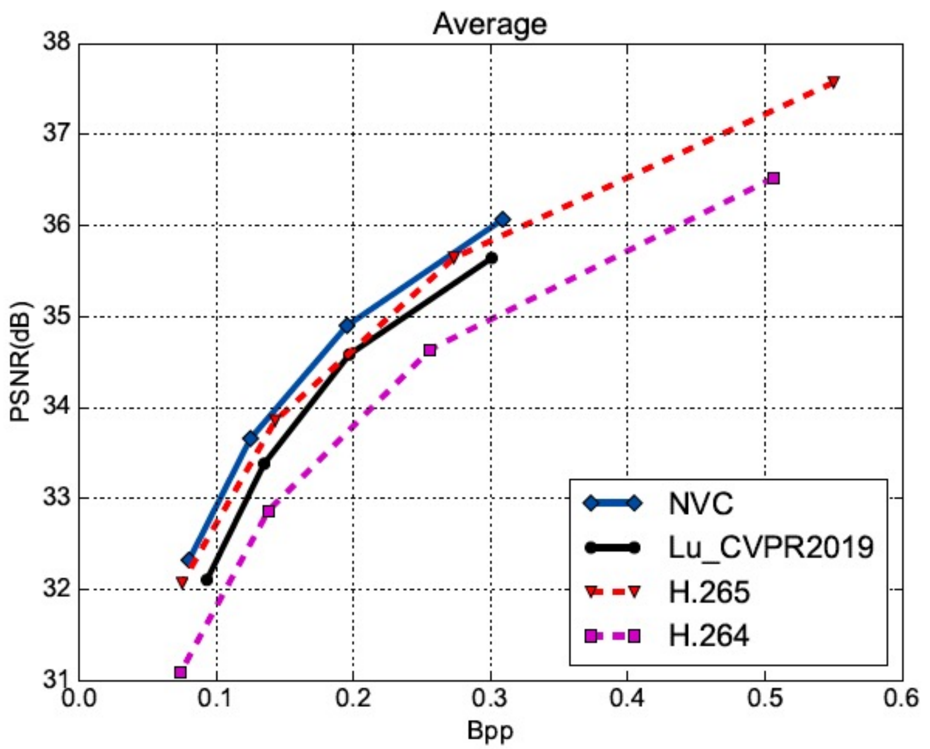
Generate and code features for the flow. Decoder generates the flow from the decoded features



Generate and code features for the flow. Decoder generates the flow in multi-resolution to enable multiscale motion compensation.

From: H. Liu *et al.*, "Neural Video Coding using Multiscale Motion Compensation and Spatiotemporal Context Model," in *IEEE Transactions on Circuits and Systems for Video Technology*, doi: 10.1109/TCSVT.2020.3035680.


overall
MS-SSIM
Bpp
0.99
0.98
0.97
0.96
0.95
0.0
0.1
0.2
0.3
0.4
0.5
0.6
NVC
Lu_CVPR2019
H.265
H.264


Average
PSNR(dB)
Bpp
38
37
36
35
34
33
32
31
0.0
0.1
0.2
0.3
0.4
0.5
0.6
NVC
Lu_CVPR2019
H.265
H.264

Recommended Readings

- Reading assignment: [Wang2002] Secs. 9.2, 9.3 (sec. 9.3.2 on OBMC optional)
- Optional Reading:
  – End-to-end Optimized Image Compression,J Ballé, V Laparra, EP Simoncelli, International Conference on Learning Representations (ICLR) 2017, *arXiv preprint arXiv:1611.01704* (2016).
  – Variational image compression with a scale hyperpriorJ Ballé, D Minnen, S Singh, SJ Hwang, N Johnston, International Conference on Learning Representations (ICLR) 2018
  – Joint autoregressive and hierarchical priors for learned image compressionD Minnen, J Ballé, GD Toderici, Advances in Neural Information Processing Systems (NeurIPS) 31, 10771-10780
  – H Liu, M Lu, Z Ma, F Wang, Z Xie, X Cao, Y Wang, "Neural Video Coding using Multiscale Motion Compensation and Spatiotemporal Context Model," in *IEEE Transactions on Circuits and Systems for Video Technology*, doi: 10.1109/TCSVT.2020.3035680.
  – D. Ding, Z. Ma, D. Chen, Q. Chen, Z. Liu and F. Zhu, "Advances in Video Compression System Using Deep Neural Network: A Review and Case Studies," in *Proceedings of the IEEE*. 2021. doi: 10.1109/JPROC.2021.3059994

Written Homework (1)

Should be able to answer all quiz questions!

1.

Consider a coding method that codes two frames as a group. Frame n is coded directly (as an I-frame), and frame n+1 is predicted from frame n and the error is coded (i.e. as a P-frame). Let a pixel in frame n be denoted by *f1*, and its corresponding pixel in frame n+1 be denoted by *f2* . f1 is directly quantized and coded, f2 is predicted from f1 using fp=f1, and the prediction error *e= f2 –fp* is quantized and coded. (for this problem, for simplicity, assume that your prediction is based on the original f1, not the quantized f1). Assume that each pixel has zero mean and variance σ^2 , and that the correlation coefficient between two corresponding pixels in two adjacent frames is ρ. Furthermore, assume the rate-distortion function for coding a single variable (original pixel or the prediction error) can be expressed as $D(R) = \varepsilon^2 \sigma^2 2^{-2R}$ where σ^2 is the variance of the variable being coded. (a) Determine the variance of the prediction error. (b) Suppose we want the average bit rate to be R (bits/pixel). How many bits you should use for *f1* and *e* respectively, to minimize the mean square error (MSE) of the reconstructed pixels? (You could assume the allocated bits can be non-integer or even negative). What would be the corresponding minimal MSE? (c) How does this method compare with coding each frame directly (as an I-frame)? Specifically, when the bit rate is the same, which one gives lower reconstruction error ?

Hint: Let R1 and R2 represent the bit rate for coding f1 and e, respectively. You should represent the quantization errors for f1 and e in terms of R1 and R2, σ^2 and ρ. You want to choose R1 and R2 so that the sum of the quantization error for both f1 and e is minimized. Because the average rate is R, you can assume R1+R2=2R, so that you only need to optimize one variable R1 or R2.

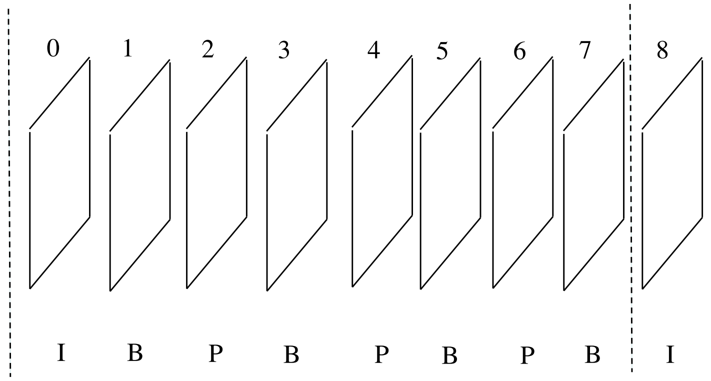
Written Homework (2)

2. Answer the following questions about I, P, B frames:

a) In video coding, a frame is often coded as either an I-frame, a P-frame, or a B-frame. Rank these modes in terms of coding efficiency and complexity, respectively. What are some of the difficulty caused by using P-frame and B-frame, in spite of their efficiency? (List one difficulty for each).

b) A video coder often divides successive frames into separate Groups of Pictures (GOP), and each GOP contains an I-frame, some P-frames and some B-frames. For the GOP structure illustrated below, what is the encoding order? What is the decoding order?

c) Assume the video frame rate is 30 frame/sec, and that encoding each frame takes 10 ms. What is the delay in millisecond at the encoder (time difference between when a frame arrives at the encoder and the time the frame is encoded) for coding an I, P, and B frames, respectively? Suppose transmitting each frame takes 100 ms, and decoding each frame takes 10 ms. What is the minimal playback delay (the time difference between when the first frame arrives at the encoder and the time the first frame should be displayed) the decoder should set, to ensure smooth display of decoded video?



Computer Assignment (OPTIONAL)

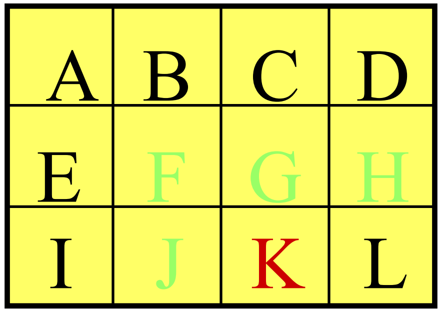
1. Write a MATLAB code that implements a basic form of the block-based hybrid video coder for coding a P-frame. For simplicity, consider only intra-prediction (using only the first 3 intra prediction modes shown in Slide ?), and unidirectional inter-prediction (with either integer or half-pel accuracy EBMA, with a search range of +/-24). You program should do the following for each 16x16 block: i) find the best intra-prediction mode and the corresponding error block and its variance; ii) find the best MV for inter-prediction and the corresponding error block and its variance; iii) Choose the prediction block whose prediction error has the smaller variance; iv) Perform 8x8 DCT on each of the 4 8x8 subblocks of the 16x16 prediction error block of the chosen mode; v) Quantize all the DCT coefficients with the same quantization stepsize q; vi) Count how many non-zero coefficients you have after quantization, vii) Reconstruct the error block by performing inverse DCT on quantized DCT coefficients; viii) Reconstruct the original block by adding the reconstructed error block into the best prediction block. Instead of developing a real entropy coder, we will use the total number of non-zero DCT coefficients as an estimate of the bit rate and ignore the bits needed to code the side information (mode info, motion vector, etc.). Your program should determine the PSNR of the reconstructed image (compared to the original image) and the total number of non-zero quantized DCT coefficients K, for a given quantization stepsize q. Apply your program to two frames of a video for several different q with a large range and determine the corresponding PSNR and K for each q. Plot PSNR vs. K as your approximate PSNR vs rate curve.   Note: if your video has very little motion, you may want to select two frames that are several frames apart).
2. Develop a MATLAB code for coding a sequence of frames, with a GOP structure of IBPBPBP. For I-frame, you should use intra-prediction only. For P-frame, you choose between intra and uni-directional inter only. For B-frame, you choose between intra, uni-directional inter, and bi-directional inter prediction. Record the average PSNR and number of non-zero DCT coefficients K for each frame type (I, P, and B) for the same q. Repeat for different q. Plot PSNR vs. K curves for different frame types as separate curves. You should observe that to achieve the same PSNR, I frame will require the highest K, followed by P, and then by B.

Additional Material

- Optimal linear predictor design
- Lagrangian multiplier for rate-distortion tradeoff
- Deadzone quantizer

Predictive Coding

- Motivation: Predicts a sample from past samples and quantize and code the error only
- Because the prediction error typically has smaller variance than the original sample, it can be represented with a lower average bit rate
- Linear predictor: use linear combination of past coded/decoded samples (in the same frame or previous frame)
- Example: linear spatial prediction



$$\hat{f}_K = a f_F + b f_G + c f_H + d f_J$$

Optimal Predictor

- Question: what predictor should we use?
  - Minimize the bit rate for coding the prediction error
  - Recall the classical relation:

$$\sigma_q^2 = \varepsilon^2 \sigma_s^2 2^{-2R}$$

  - Because quantization error with a given bit rate depends on the variance of the signal,
    - minimizing the quantization error = minimizing the prediction error variance (=mean square prediction error).

Linear Minimal MSE Predictor

- Linear Predictor:

$$s_p = \sum_{k=1}^{K} a_k s_k$$

- Prediction error:

$$\sigma_p^2 = E\{|\mathcal{S}_0 - \mathcal{S}_p|^2\} = E\left\{\left|\mathcal{S}_0 - \sum_{k=1}^{K} a_k \mathcal{S}_k\right|^2\right\}.$$

- Optimal coefficients must satisfy (by setting derivative with respect to a_k to zero):

$$E\left\{\left(\mathcal{S}_0 - \sum_{k=1}^{K} a_k \mathcal{S}_k\right)\mathcal{S}_l\right\} = 0, \quad l = 1, 2 \ldots, K.$$

$$\sum_{k=1}^{K} a_k R(k, l) = R(0, l), \quad l = 1, 2 \ldots, K,$$

- Useful if the signal is stationary and has known correlations among samples!

Linear Minimal MSE Predictor

- Linear Predictor: $$s_p = \sum_{k=1}^{K} a_k s_k$$

- Prediction error: $$\sigma_p^2 = E\{|\mathcal{S}_0 - \mathcal{S}_p|^2\} = E\left\{\left|\mathcal{S}_0 - \sum_{k=1}^{K} a_k \mathcal{S}_k\right|^2\right\}.$$

- Optimal coefficients must satisfy (by setting derivative with respect to a_k to zero):
$$E\left\{\left(\mathcal{S}_0 - \sum_{k=1}^{K} a_k \mathcal{S}_k\right)\mathcal{S}_l\right\} = 0, \quad l = 1, 2 \ldots, K.$$
$$\sum_{k=1}^{K} a_k R(k, l) = R(0, l), \quad l = 1, 2 \ldots, K,$$

- Useful if the signal is stationary and has known correlations among samples!

Matrix Form

- The previous equation can be rewritten as:

$$\begin{bmatrix} R(1,1) & R(2,1) & \cdots & R(K,1) \\ R(1,2) & R(2,2) & \cdots & R(K,2) \\ \cdots & \cdots & \cdots & \cdots \\ R(1,K) & R(2,K) & \cdots & R(K,K) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \cdots \\ a_K \end{bmatrix} = \begin{bmatrix} R(0,1) \\ R(0,2) \\ \cdots \\ R(0,K) \end{bmatrix}$$

$$[\mathbf{R}]\mathbf{a} = \mathbf{r}.$$

$$\mathbf{a} = [\mathbf{R}]^{-1}\mathbf{r}.$$

$$\sigma_p^2 = E\{(\mathcal{S}_0 - \mathcal{S}_p)\mathcal{S}_0\} = R(0,0) - \sum_{k=0}^{K} a_k R(k,0)$$

$$= R(0,0) - \mathbf{r}^T\mathbf{a} = R(0,0) - \mathbf{r}^T R^{-1}\mathbf{r}.$$

- R(k,l) is the correlation between k-th and l-th prediction sample. R(k,0) is the correlation of the current sample with the k-th prediction sample.

Predictive Coding Gain

Recall: Reconstruction error for original samples = quantization error of prediction errors.

$D_{DPCM}=D_q(R)$

Distortion-rate for predictive coder (DPCM)

$$D_{\text{DPCM}} = \epsilon_p^2 \sigma_p^2 2^{-2R}$$

Distortion-rate for coding the samples directly (PCM)

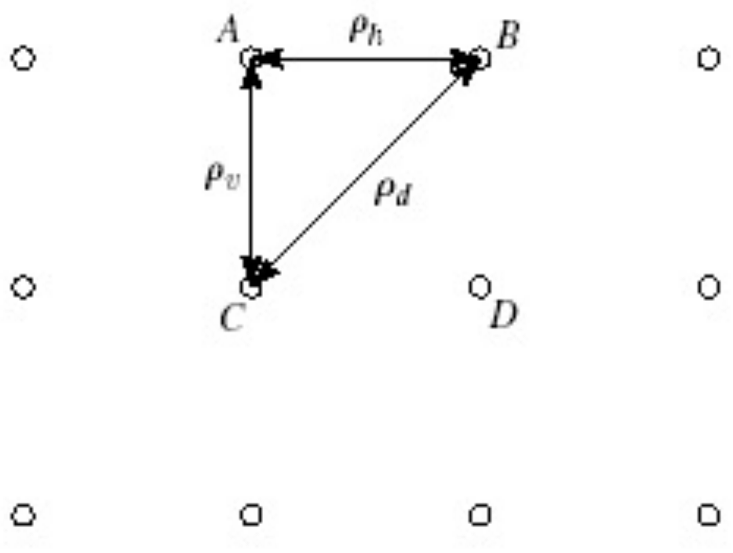
$$D_{PCM} = \varepsilon_s^2 \sigma_s^2 2^{-2R}$$

Coding gain of DPCM:

$$G_{\text{DPCM}} = \frac{D_{\text{PCM}}}{D_{\text{DPCM}}} = \frac{\epsilon_s^2}{\epsilon_p^2} \frac{\sigma_s^2}{\sigma_p^2}$$

PCM: Pulse coded modulation (quantize original sample directly)
DPCM: Differential PCM (quantize the difference of original sample and predicted sample)

Example



$$\hat{D} = a_1 C + a_2 B + a_3 A$$

Example Continued

$$\begin{bmatrix} R(C,C) & R(C,B) & R(C,A) \\ R(B,C) & R(B,B) & R(B,A) \\ R(A,C) & R(A,B) & R(A,A) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} R(D,C) \\ R(D,B) \\ R(D,A) \end{bmatrix}$$

$$\begin{bmatrix} 1 & \rho_d & \rho_v \\ \rho_d & 1 & \rho_h \\ \rho_v & \rho_h & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} \rho_h \\ \rho_v \\ \rho_d \end{bmatrix}.$$

In the special case of $\rho_h = \rho_v = \rho$, $\rho_d = \rho^2$, the optimal predictor is

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} \rho \\ \rho \\ -\rho^2 \end{bmatrix}.$$

The MSE of this predictor, using Equation (9.2.10), is

$$\sigma_p^2 = R(0,0) - [R(0,1) \quad R(0,2) \quad R(0,3)] \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = (1-\rho^2)^2 \sigma_s^2.$$

$$G_{\mathrm{DPCM}} = \frac{\sigma_s^2}{\sigma_p^2} = \frac{1}{(1-\rho^2)^2}$$

(DPCM is better than TC for this case!)

How to Choose the Lagrange Parameter? (not required)

$L(m) = D(m) + \lambda R(m)$

Think of mode m as a continuous variable, m* should be chosen so that

$\frac{\partial L}{\partial m} = \frac{\partial D}{\partial m} + \lambda \frac{\partial R}{\partial m} = 0 \text{ or } \lambda = -\frac{\partial D}{\partial m} / \frac{\partial R}{\partial m} = -\frac{\partial D}{\partial R}$

λ = Negative distortion rate slope at the operating rate or distortion (depending on q)!

Distortion and rate depends on quantization error.
Assume uniform quantization with stepsize q .
When q is small, we can assume

$$D(q) = \frac{q^2}{12}$$

Furthermore $D(R) = b2^{-aR}$ or $R = -\frac{1}{a} log_2(\frac{D}{b}) = -\frac{1}{a} log_2(\frac{q^2}{12b})$

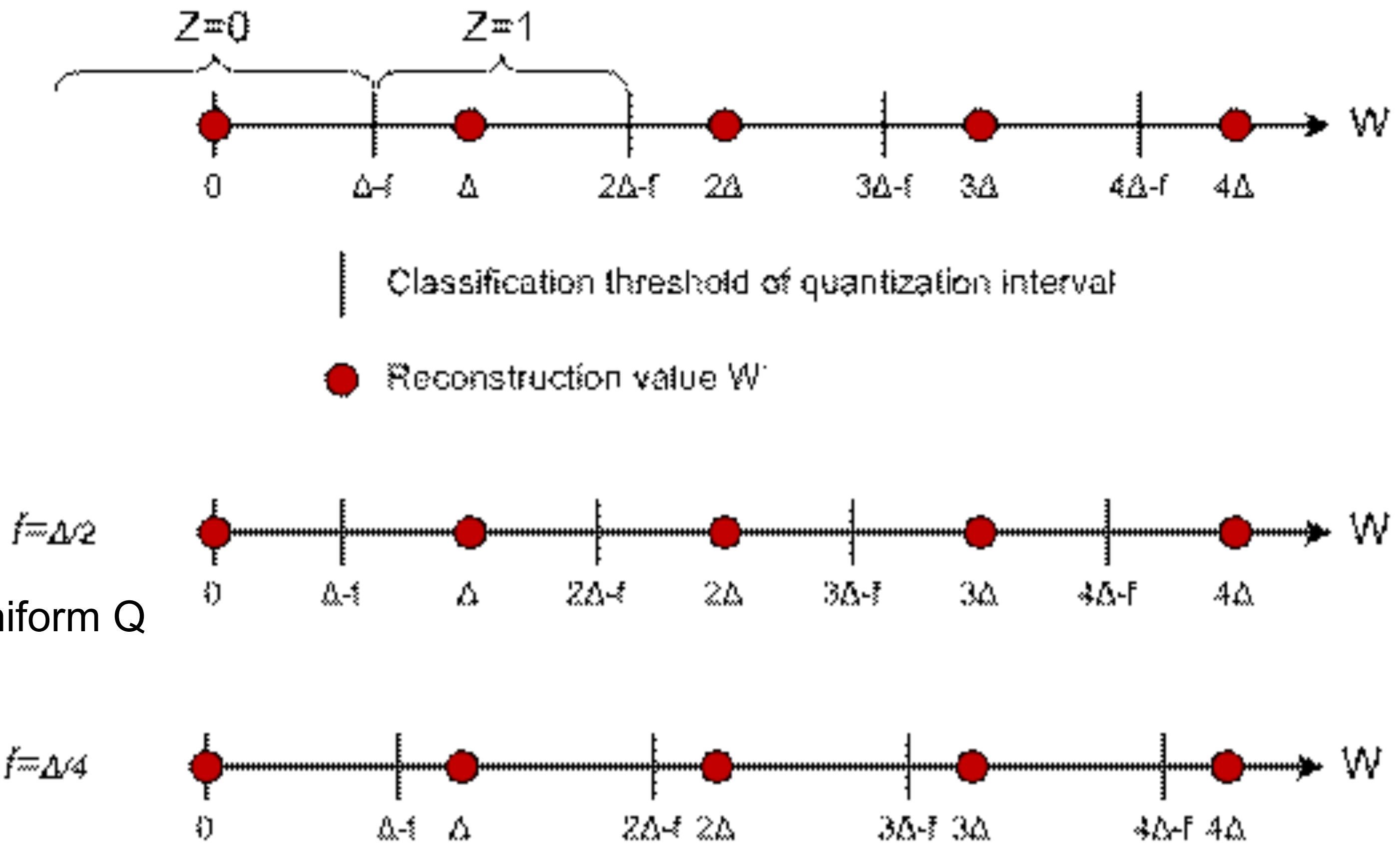
Therefore:

$\frac{\partial D}{\partial q} = \frac{1}{6} q, \frac{\partial R}{\partial q} = -\frac{24b}{a\, ln2\, q}, \lambda = -\frac{\partial D}{\partial R} = -\frac{\partial D}{\partial q} / \frac{\partial R}{\partial q} = cq^2$

λ should be proportional to q^2!

- Practical Solution: Determine c experimentally!
- T. Wiegand and B. Girod, "LAGRANGE MULTIPLIER SELECTION IN HYBRID VIDEO CODER CONTROL," Proc. ICIP2001

Deadzone Quantizer for Transform Coefficients



Uniform Q

From http://www.h265.net/2009/06/quantization-techniques-in-jmkta-part-2.html

Why deadzone quantizer?

- Recall the “centroid condition” – Reconstruction level = centroid (conditional mean) of a partition interval
- For non-uniform symmetric pdf, it is expected that the reconstruction level would not be in the middle of an interval
- Transform coefficients of prediction error can be modeled well by Laplacian distribution

$$p(w)=\frac{\lambda}{2}\exp\{-\lambda|w|\}$$

Centroid in the interval $(k\Delta - f, (k+1)\Delta - f)$:

$$W_k=\int_{k\Delta-f}^{(k+1)\Delta-f} wp(w)dw \Big/ \int_{k\Delta-f}^{(k+1)\Delta-f} p(w)dw$$

$$=k\Delta - f+\frac{1}{\lambda}-\frac{\Delta}{e^{\lambda\Delta}-1}$$

Let $W_k = k\Delta$, we get

$$f^*=\frac{1}{\lambda}-\frac{\Delta}{e^{\lambda\Delta}-1}$$

For Laplacian source, by choosing the particular partition structure of deadzone quantizer and set “f” in this way, the reconstruction level in each interval minimizes the quantization error of that interval!

For H.264: f=Δ/3 for intra modes,
f=Δ/6 for Inter modes (with more peaky distribution or larger λ)

$$p(w) = \frac{\lambda}{2}\exp\{-\lambda|w|\}$$

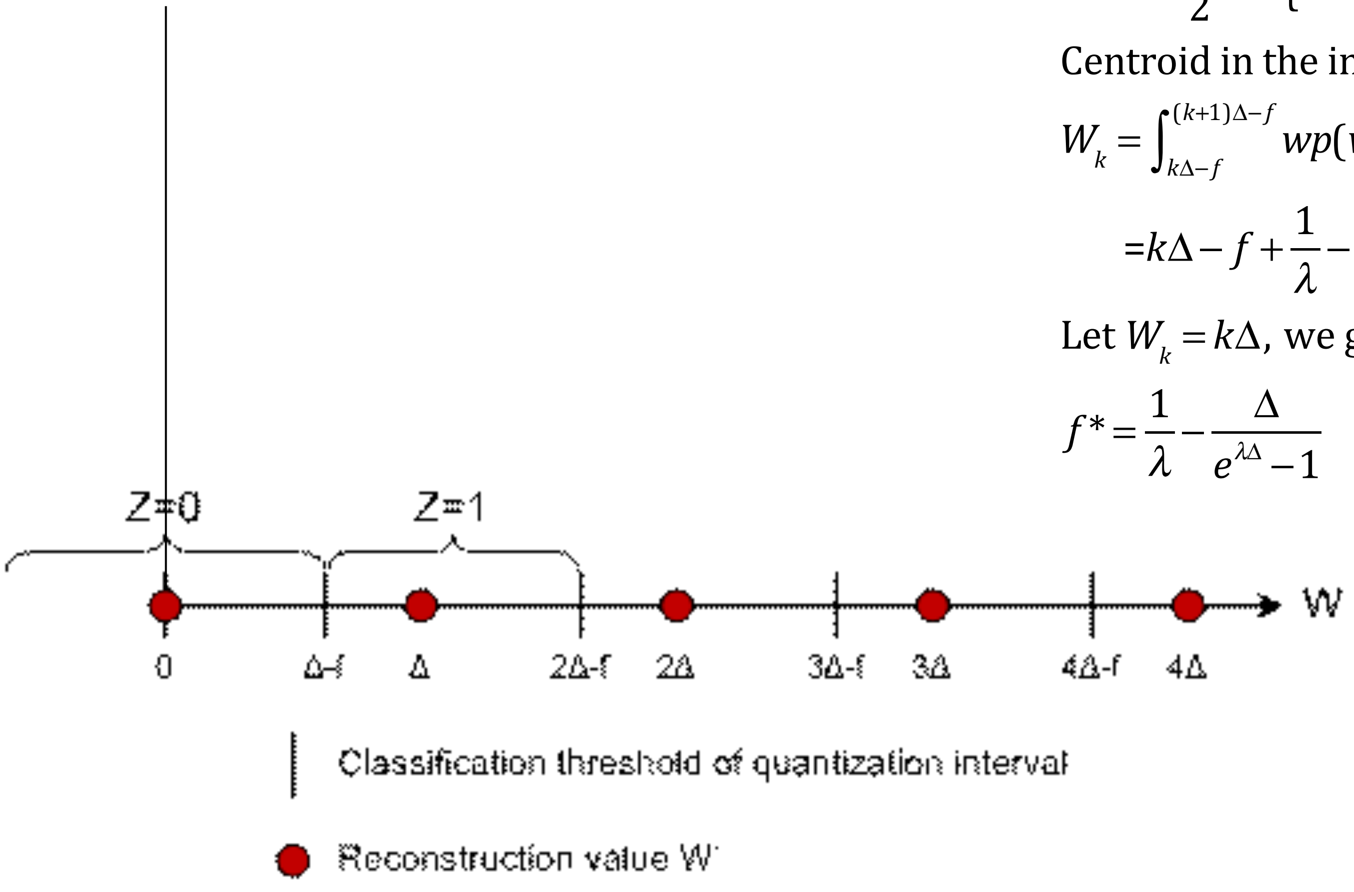
Centroid in the interval $(k\Delta - f, (k+1)\Delta - f)$:

$$W_k = \int_{k\Delta - f}^{(k+1)\Delta - f} wp(w)dw \Big/ \int_{k\Delta - f}^{(k+1)\Delta - f} p(w)dw$$

$$= k\Delta - f + \frac{1}{\lambda} - \frac{\Delta}{e^{\lambda\Delta} - 1}$$

Let $W_k = k\Delta$, we get

$$f^* = \frac{1}{\lambda} - \frac{\Delta}{e^{\lambda\Delta} - 1}$$



Z=0
Z=1
W
0
Δ-f
Δ
2Δ-f
2Δ
3Δ-f
3Δ
4Δ-f
4Δ
Classification threshold of quantization interval
Reconstruction value W'