Distributed Source Coding for Image and Video Applications

Ngai-Man (Man) CHEUNG

Signal and Image Processing Institute
University of Southern California

http://biron.usc.edu/~ncheung/

Acknowledgements

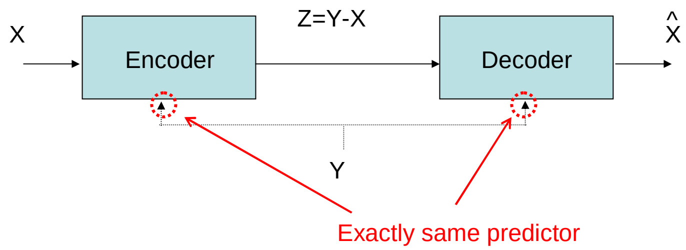
- Collaborators at USC:
  - Antonio Ortega
  - Huisheng Wang
  - Caimu Tang
  - Ivy Tseng
- Some materials taken from literatures:
  - Xiong et al (Texas A&M)
  - Girod et al (Stanford)
  - Ramchandran et al (UC Berkeley)
  - Others

Outline

- Introduction
  - Motivation
  - Source coding with decoder side-information only
  - Simple example to illustrate DSC idea
  - Application scenarios
- Basic information-theoretic results
  - Slepian-Wolf
  - Wyner-Ziv
- Practical encoding/decoding algorithm
  - Role
  - LDPC based
- Applications
  - Low-complexity video encoding
  - Scalable video coding
  - Flexible decoding, e.g., multiview video

Motivation

Compression with Closed-loop Prediction (CLP): E.g., MPEG, H.26X

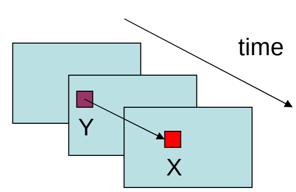


- Compute and send the difference/prediction residue (Z=Y-X) to decoder
- Predictor Y:
  - Motion-compensated predictor in neighboring frames
  - Co-located pixels in neighboring slices in volumetric image

5

Issues in Closed-loop Prediction: High Complexity Encoding

- High complexity encoding
  - Motion estimation



- Some emerging applications require low complexity encoding, e.g., mobile video, video sensors

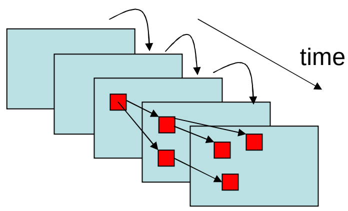
Distributed source coding allows flexible distribution of complexity between encoder and decoder

6

Issues in Closed-loop Prediction: Vulnerable to Transmission Error

USC Viterbi School of Engineering

- Error in predictor propagates to subsequent frames: drifting



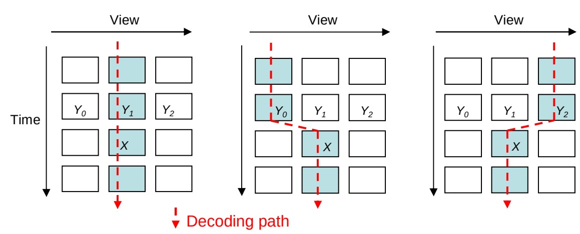
Distributed source coding allows exact reconstruction with transmission error

USC

Issues in Closed-loop Prediction: Lack of Decoding Flexibility

USC Viterbi School of Engineering

- Some emerging applications need to support multiple decoding paths, e.g., multiview video



- When users can choose among different decoding paths, it is not clear which previous reconstructed frame will be available to use in the decoding

DSC can support multiple decoding paths and address predictor uncertainty efficiently

USC

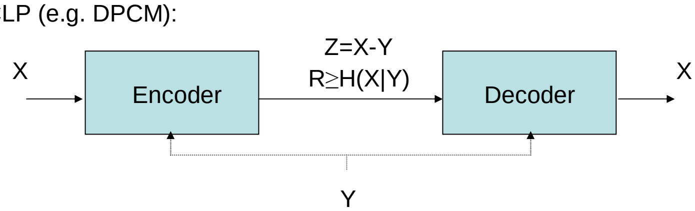
Distributed Source Coding

Distributed Source Coding

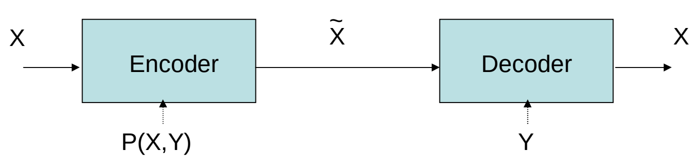
Lossless compression of random variable X

- Intra coding: $R \geq H(X)$

(i) CLP (e.g. DPCM):



(ii) DSC:



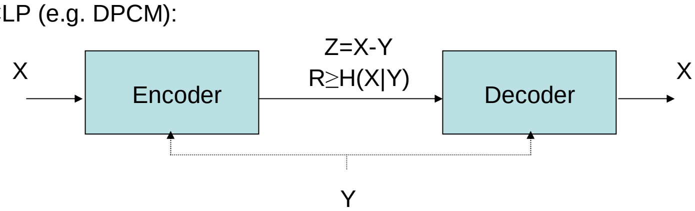
- **Encoding does not require Y**
  - Y de-coupled from the encoded data $\tilde{X}$

Distributed Source Coding

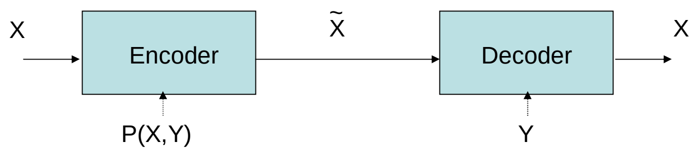
USC Viterbi School of Engineering

Lossless compression of random variable X
- Intra coding: $R \geq H(X)$

(i) CLP (e.g. DPCM):



(ii) DSC:



- **Encoding does not require Y**
  - Y de-coupled from the encoded data $\tilde{X}$

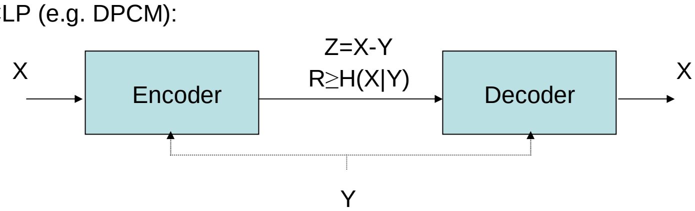
USC

Distributed Source Coding

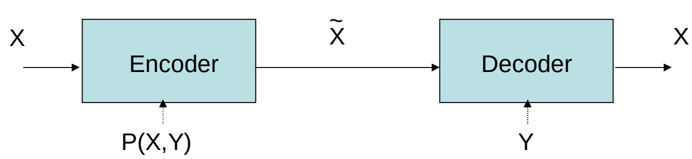
USC Viterbi School of Engineering

Lossless compression of random variable X
- Intra coding: $R \geq H(X)$

(i) CLP (e.g. DPCM):



(ii) DSC:



- **Encoding does not require Y**
  - Y de-coupled from the encoded data $\tilde{X}$

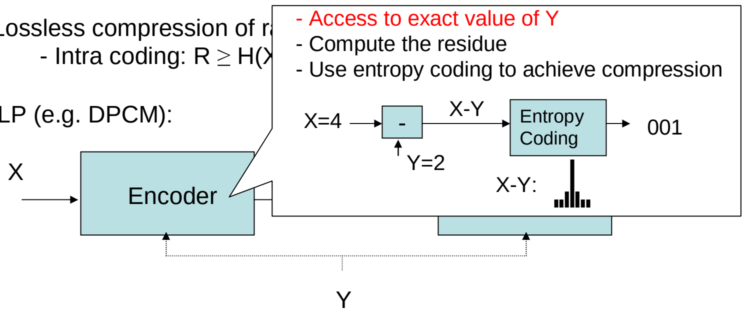
USC

Distributed Source Coding

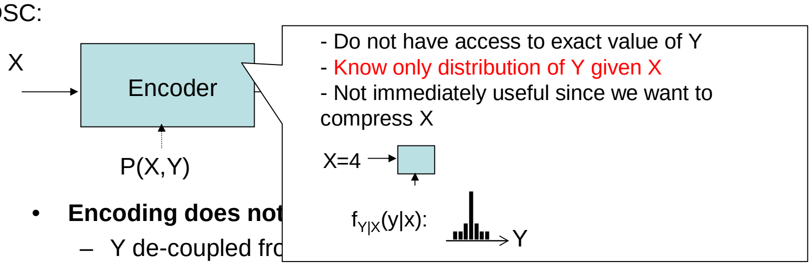
USC Viterbi School of Engineering

Lossless compression of r
- Intra coding: R ≥ H(X

(i) CLP (e.g. DPCM):



(ii) DSC:



- **Encoding does not**
  - Y de-coupled fro

USC

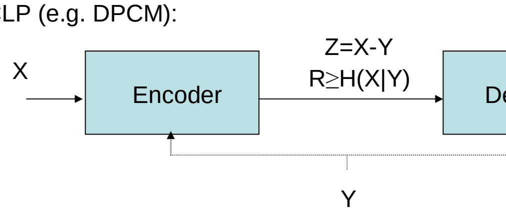
13

Distributed Source Coding

USC Viterbi School of Engineering

Lossless compression of random variable X
- Intra coding: R ≥ H(X)

(i) CLP (e.g. DPCM):

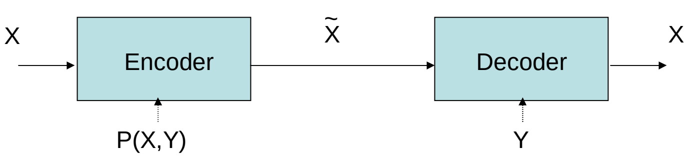


-Why DSC? What are the potential applications?

-How to encode the input exactly?

-What is the coding performance?

(ii) DSC:



- **Encoding does not require Y**
  - Y de-coupled from the encoded data $\tilde{X}$

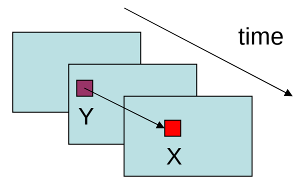
USC

14

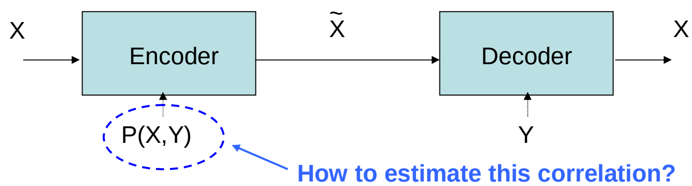
Why DSC?
When predictor is not available during encoding

USC Viterbi School of Engineering

- Correlated sources are not co-located
  - Sensors network, wireless camera
- Low complexity video encoding
  - Due to complexity constraint at the encoder

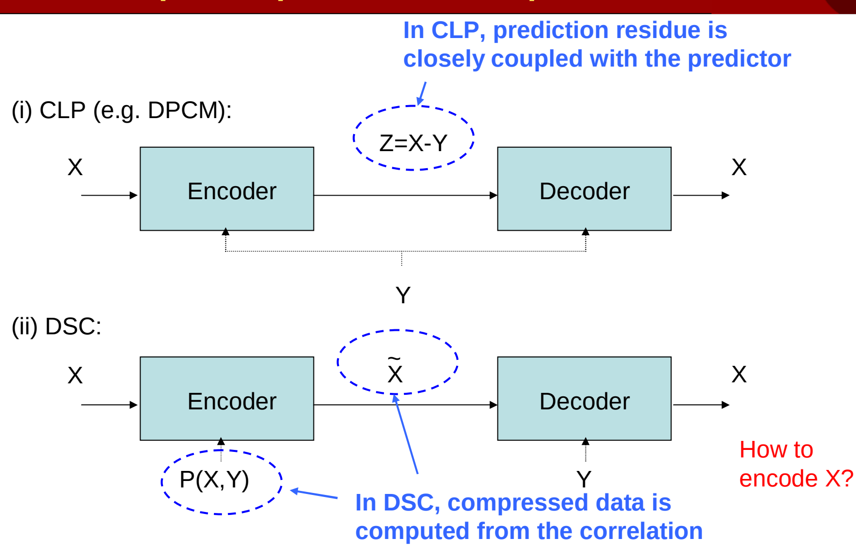


DSC:



Why DSC?
To “decouple” compressed data from predictor

USC Viterbi School of Engineering

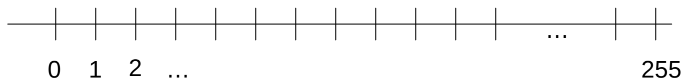


More robust to uncertainty on or error in Y

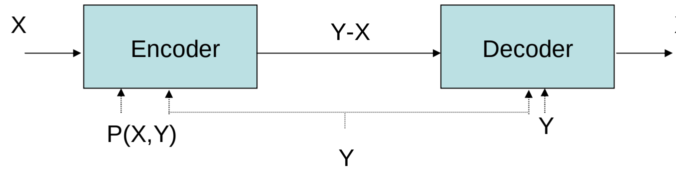
DSC – Example Using Coset

USC Viterbi School of Engineering

- X takes value [0, 255] (uniformly)
  - Intra coding requires 8 bits for lossless



- Correlation P(X,Y): 4 > Y-X ≥ -4 (i.e., Y-X takes value in [-4,4))
  - Can explore this correlation
- If Y is available at both the encoder and decoder
  - CLP: communicate the residue



Requires 3 bits to convey X

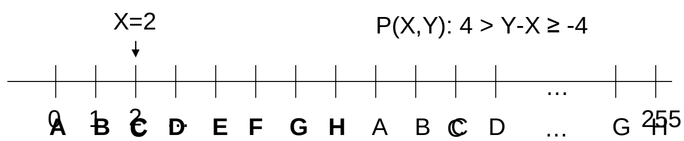
If Y is not available at the encoder, how can we communicate X?

USC

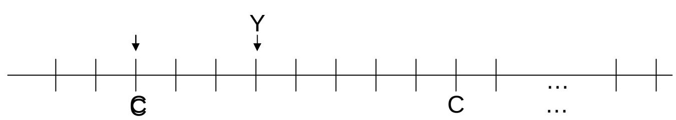
DSC – Example Using Coset (Cont'd)

USC Viterbi School of Engineering

- How to communicate X if Y is not available at the encoder?



- **Partition** reconstruction levels into different cosets
  - Each coset includes several reconstruction levels
- Encoder: **transmit coset label** (syndrome), need 3 bits
- Decoder: **disambiguate** label (closest to Y)

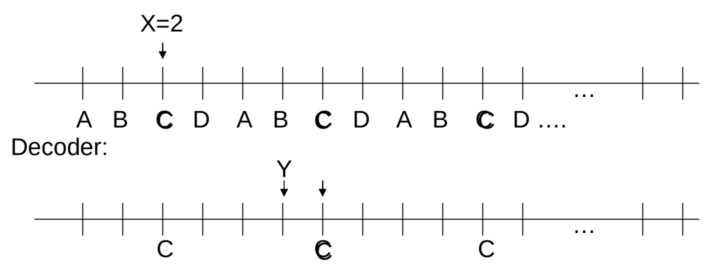


Requires 3 bits to convey X

USC

DSC – Example Using Coset (Cont'd)

- Can we use less than 3 bits?
- Try 2 bits



Decoding error if we use less bits than required

- Correlation information determines the number of coset for error-free reconstruction
- If correlation were 2 > Y-X ≥ -2 (i.e., X, Y are more correlated), 2 bits would work

DSC - Main Ideas

- Encoding: Send **ambiguous** information to achieve compression
  - E.g., one label represents group of reconstruction levels
- Decoding: information is **disambiguated** at the decoder using side information Y
  - Pick the coset member closest to Y
- "Level of ambiguity" (hence the encoding rate) is determined by correlation between X and Y
  - More correlated, more ambiguous representation using less bits

DSC - Main Steps

USC Viterbi School of Engineering

- Partition input into cosets
- Members in the coset are separated by minimum distance
- Send coset index
- At decoder, pick the member closest to side information in the coset
- **Similar steps for advanced algorithm based on error correction code (E.g., LDPC)**

How about the coding efficiency?

USC

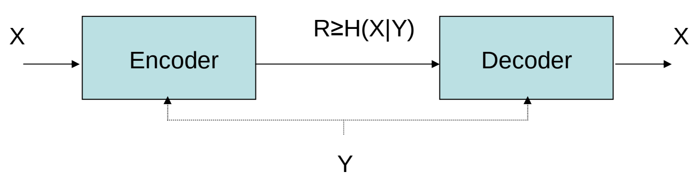
21

DSC – Theoretical Performance

USC Viterbi School of Engineering

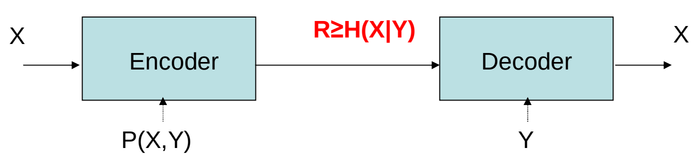
Lossless compression of random variable X

- CLP



- DSC

[Slepian, Wolf; 1973]



Efficient encoding possible even when encoder does not have precise knowledge of Y

USC

22

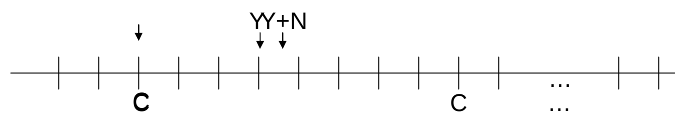
DSC Properties - Inherent Error Resilience

USC Viterbi School of Engineering

- Even Y is corrupted by noise, it is still possible to reconstruct X exactly
  - Prevent error propagation

Encoder:

Decoder:



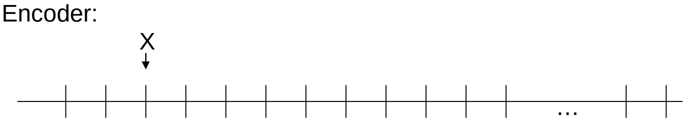
Compressed data (i.e., coset label) is decoupled from Y

DSC Properties – Robust to Predictor Uncertainty

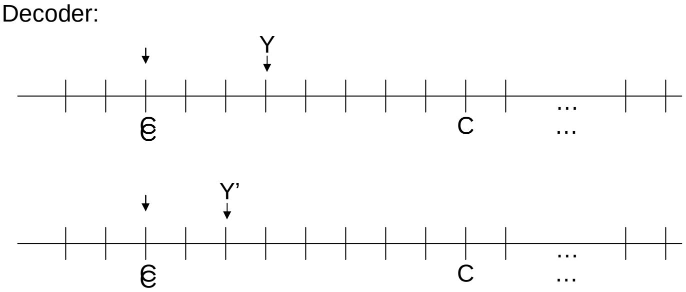
USC Viterbi School of Engineering

- Multiple predictor candidates at the decoder (e.g., multiview video)
  - Encoder does not know exactly which predictor is available at decoder

Encoder:



Decoder:



USC Viterbi School of Engineering

Application Scenario

USC

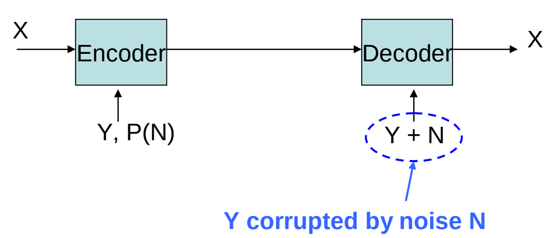
25

Application Scenario – Video Transmission

USC Viterbi School of Engineering

- Resilience to transmission error

- In practical applications, we have to estimate the noise power at the encoder



[Sehgal, Jagmohan, Ahuja; IEEE Trans. Multimedia 04],
[Majumdar, Wang, Ramchandran, Garudadri; PCS 04],
[Rane, Girod; VCIP 06], …

USC

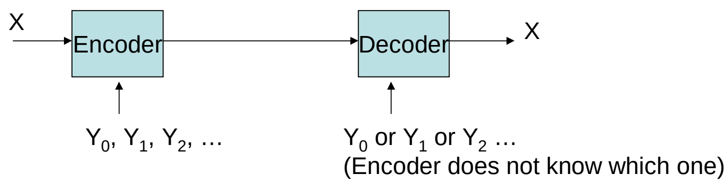
26

Application Scenario – Flexible Decoding

USC Viterbi School of Engineering

Flexible decoding, e.g., multiview video

- DSC is robust to predictor uncertainty
- Encoder estimates correlation between X and Y_k
- Recover X exactly no matter which Y_k is at decoder



[Cheung, Ortega; MMSP 07]

USC

Application Scenario – Low Complexity Video Encoding

USC Viterbi School of Engineering

Low complexity video encoding

- Finding Y to predict X is complex
- P(X,Y) can be estimated in low complexity, hopefully…

[Puri, Ramchandran, Allerton 02]
[Aaron, Zhang, Girod, Asilomar 02], …

USC

Outline

USC Viterbi School of Engineering

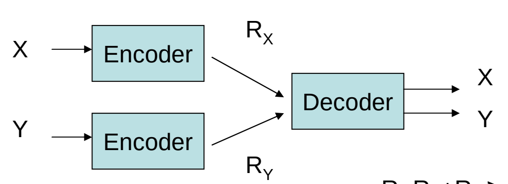
- Introduction
  - Motivation
  - Source coding with decoder side-information only
  - Simple example to illustrate DSC idea
  - Application scenarios
- Basic information-theoretic results
  - Slepian-Wolf
  - Wyner-Ziv
- Practical encoding/decoding algorithm
  - Role
  - LDPC based
- Applications
  - Low-complexity video encoding
  - Scalable video coding
  - Flexible decoding, e.g., multiview video

USC

Slepian-Wolf Theorem

USC Viterbi School of Engineering

- Lossless compression of correlated sources: **X**={X}, **Y**={Y}
- Random variables X, Y jointly distributed P(X,Y)
- **Possible to achieve total rate as low as H(X,Y) - theoretical limit when the encoders can cooperate**
- Decoder side-information is a special case



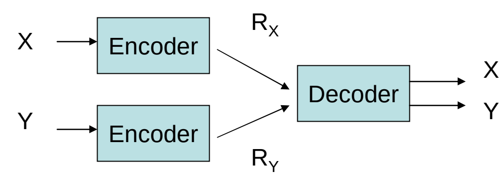
$R=R_X+R_Y \geq H(X,Y)$

Distributed source coding with decoder side-information:

$R_Y \geq H(Y)$
$R_X \geq H(X|Y)$
$R \geq H(Y)+H(X|Y)=H(X,Y)$

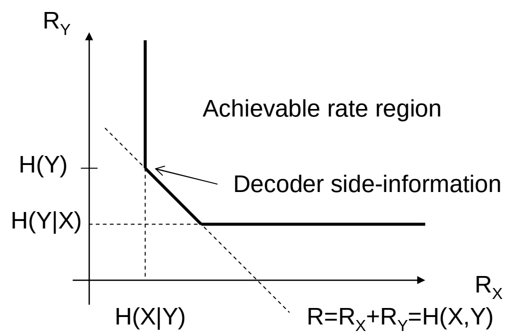
USC

Slepian-Wolf Theorem (cont')



General case:

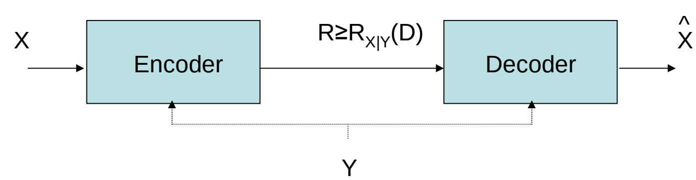
$R_Y \geq H(Y|X)$
$R_X \geq H(X|Y)$
$R \ \geq H(X,Y)$

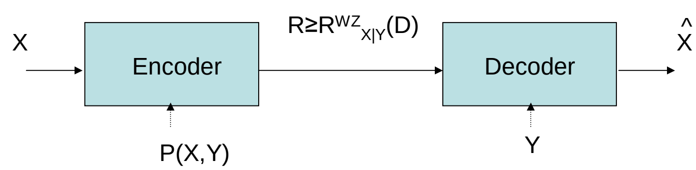


For video/image applications, usually operate at corner points (decoder side-information)

Wyner-Ziv Theorem

- Lossy compression of **X**={X}, **Y**={Y}: memoryless sources
- Y is available only at the decoder





In general, there is rate loss: $R^{WZ}_{X|Y}(D)-R_{X|Y}(D) \geq 0$

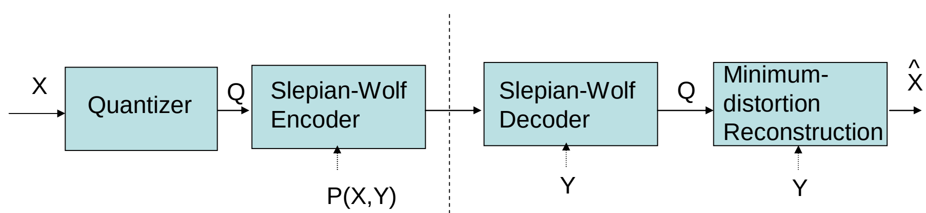
Wyner-Ziv Theorem – Example

- Example: quadratic Gaussian case
- **X**={X}, **Y**={Y}; X, Y are jointly Gaussian
- Distortion metric is MSE
- No rate loss with Wyner-Ziv coding
- $R^{WZ}_{X|Y}(D) = R_{X|Y}(D)$

- No rate loss in several other cases

Practical Wyner-Ziv Coding

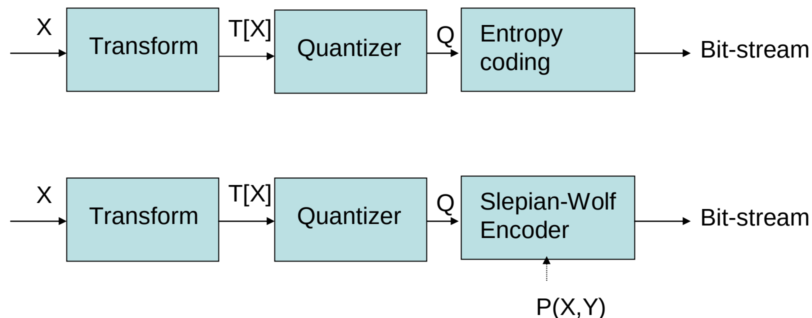
- Practical Wyner-Ziv coding:



- Lossy qunatization
- Lossless compression of quantization index Q using SW encoder
- At decoder, side information Y is used in:
  - Slepian-Wolf decoding
  - Reconstruct X in the quantization bin specified by Q
- To achieve Wyner-Ziv limit, need
  - Good quantization: E.g., TCQ
  - Good Slepian-Wolf code: E.g., based on LDPC

Practical Slepian-Wolf Code Construction

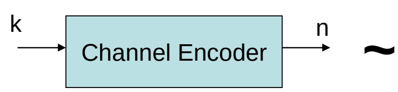
- **Slepian-Wolf encoder plays a similar role as entropy coding in conventional source coding problem**



- Lossless compression
- Rely on transform, quantization to achieve good (lossy) coding performance

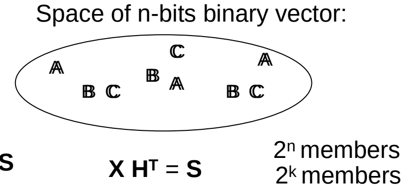
SW Coding with Error Correcting Code

- Linear (n, k) binary error-correcting code
  - k bits input to channel encoder
  - n bits output at channel encoder
  - n-k bits redundancy



- Defined by parity check matrix **H**

Space of n-bits binary vector:



- For any n-bits binary vector **X**
  - **X H**^{T} = **S, S** is syndrome (n-k bits vector)
  - If **X** is one of the 2^k codeword, **S**=**0**
  - In general, 2^k different **X** have the same **S**
  - Partition space of **X** into 2^{n-k} cosets
  - Each coset has 2^k members
  - In each coset, the minimum (Hamming) distance between the members is the same

LDPC Based Slepian-Wolf Code

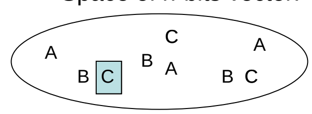
USC Viterbi School of Engineering

- **X**={X}, **Y**={Y}, to compress a n-bits binary vector **X**
- **Similar idea as the coset example we have mentioned**

Encoder:

- Partition input space into cosets:
- Send coset index:

Space of n-bits vector:



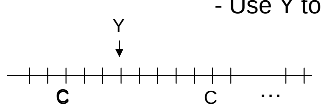
X mod m = coset index

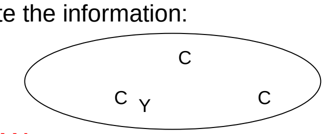
2^{n-k} cosets

$\mathbf{X\,H^T = S}$

Decoder:

- Use Y to disambiguate the information:





- Compression performance: (n-k)/n

- Number of cosets (min. distance between members of coset) depends on correlation

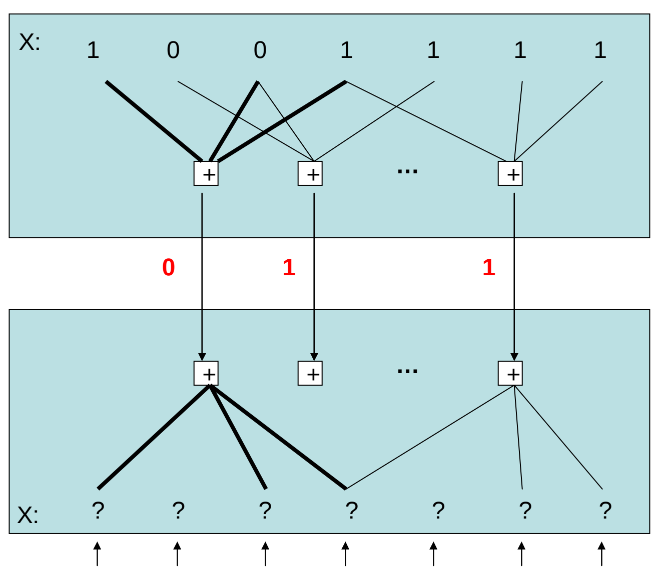
LDPC Based Slepian-Wolf Code (cont')

USC Viterbi School of Engineering

SW Encoder:

$X\,H^T = S$

SW Decoder:



$p(X_i \mid Y_i)$

Decoder would use the SI to estimate the probability that each bit is being zero or one

Outline

- Introduction
  - Motivation
  - Source coding with decoder side-information only
  - Simple example to illustrate DSC idea
  - Application scenarios
- Basic information-theoretic results
  - Slepian-Wolf
  - Wyner-Ziv
- Practical encoding/decoding algorithm
  - Role
  - LDPC based
- Applications
  - Low-complexity video encoding
  - Scalable video coding
  - Flexible decoding, e.g., multiview video

Low Complexity Video Encoding

- Shift motion estimation to decoder
- Low complexity video encoder
- High complexity decoder

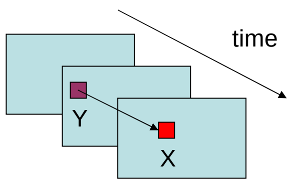
- Application: uplink video transmission
  - Video sensor network
  - Mobile phone video

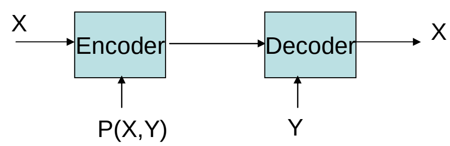
- Important work:
  - [Puri, Ramchandran; 2002]
  - [Aaron, Zhang, Girod; 2002]

Low Complexity Video Encoding - Example

USC Viterbi School of Engineering

- [Puri, Ramchandran; 2002], [Puri, Majumdar, Ramchandran; 2007]
- DSC requires only P(X,Y) at encoder
  - X: current macroblock to be encoded
  - Y: best motion-compensated predictor from reference frame for X
- Two problems:
  - At encoder, how to estimate P(X,Y) without searching Y?
    - For low complexity encoding, avoid ME at encoder
  - At decoder, how to obtain Y to decode X?
    - Conventional ME requires X to locate Y
    - X is not available



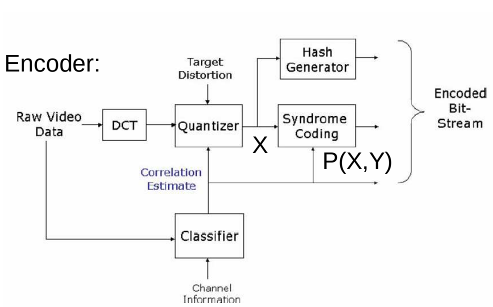


USC

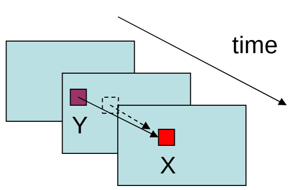
Low Complexity Video Encoding – Example (cont')

USC Viterbi School of Engineering

Encoder:



- Block based
- Transform, Quantization, Syndrome coding
- Estimate P(X,Y):
  - Squared difference (SD) between co-located block in previous frame
  - Use SD to infer P(X,Y) thro' classifier
  - Require no motion search



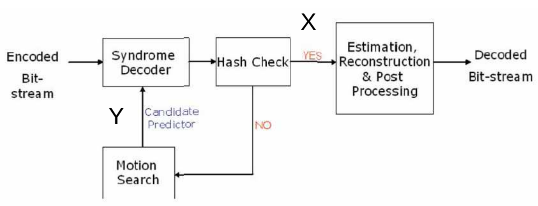
[Puri, Majumdar, Ramchandran; IEEE Trans. IP 2007]

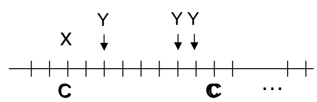
USC

Low Complexity Video Encoding – Example (cont')

USC Viterbi School of Engineering

Decoder:





- Require Y (best motion-compensated predictor) to perform Slepian-Wolf decoding
- Conventional ME requires X to locate Y
- But X is not available
  - Encoder sends also CRC of X
  - At decoder, try all possible candidate predictors
  - For each candidate predictor, SW decoder outputs one decoded sequence closest to the candidate
  - If the decoded sequence passes CRC test, declare current decoded sequence as X
  - Otherwise, try another candidate

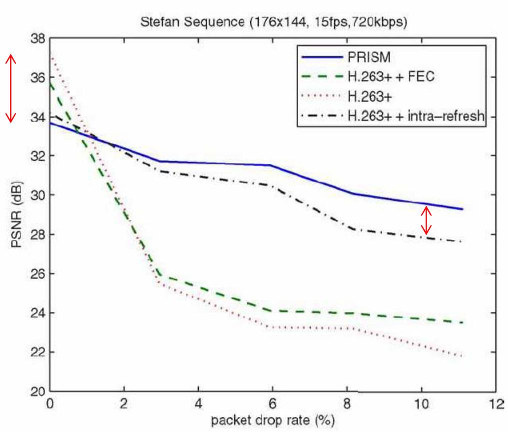
USC

Low Complexity Video Encoding – Coding Efficiency

USC Viterbi School of Engineering

- Compared to H.263+, FEC, Intra-refresh

3dB loss when no packet drop



2dB gain at 10% packet drop rate

[Puri, Majumdar, Ramchandran; IEEE Trans. IP 2007]

USC

USC Viterbi School of Engineering

DSC Application to Scalable Video Coding

USC

Some Work on Scalable Coding Based on DSC

USC Viterbi School of Engineering

- [Xu, Xiong; VCIP 04, IEEE Trans. IP 06]
  - Embedded enhancement layer similar to MPEG-4/H.26L FGS
  - Robust to error in base layer
  - Do not suffer performance loss due to layering
- [Tagliasacchi, Majumdar, Ramchandran, Tubaro; PCS 04, Eurasip SP 06]
  - Spatial/temporal/SNR scalability
  - Based on PRISM [Puri, Ramchandran; Allerton 02]
  - Robust to channel losses
  - Flexible distribution of complexity
- [Sehgal, Jagmohan, Ahuja; PCS 04]
  - Multiple Wyner-Ziv encoded versions for different possible predictors
  - Encoder streams an appropriate encoded version based on knowledge of predictors available at decoder
- [Wang, Cheung, Ortega; PCS 04, Eurasip JASP 06]
  - Improvement of MPEG4-FGS: Utilizing EL reconstruction for prediction
  - Low complexity overhead: Avoid replicating EL reconstruction at encoder

USC

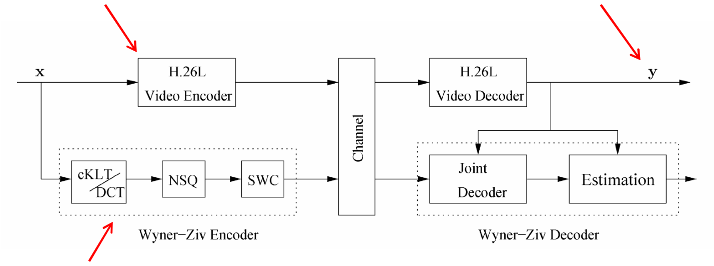
Error Robust Scalable Coding

USC Viterbi School of Engineering

[Xu, Xiong; VCIP 04, IEEE Trans. IP 06]

Base layer (BL): Standard video codec

Side Information (SI): BL reconstruction



Enhancement layer (EL) based on bit-plane coding and DSC

Error corrupted base layer (SI) may still be used for DSC decoding

USC

50

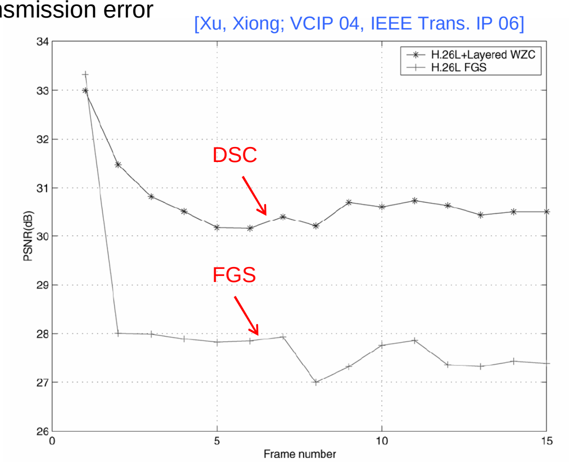
Robust to Transmission Error

USC Viterbi School of Engineering

DSC-based system can achieve substantial improvement in situations with transmission error

[Xu, Xiong; VCIP 04, IEEE Trans. IP 06]

Football
1% macroblock loss at BL
BL: 1450 kbps
EL: 200 kbps



In addition, the system does not suffer performance loss due to layering, i.e., same performance as the monolithic WZ coding

USC

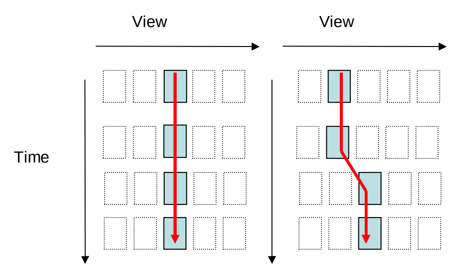
51

DSC Application to Flexible Video Decoding

Flexible Video Decoding

Apply DSC to generate a *single* bitstream that can be decoded in *several* different ways

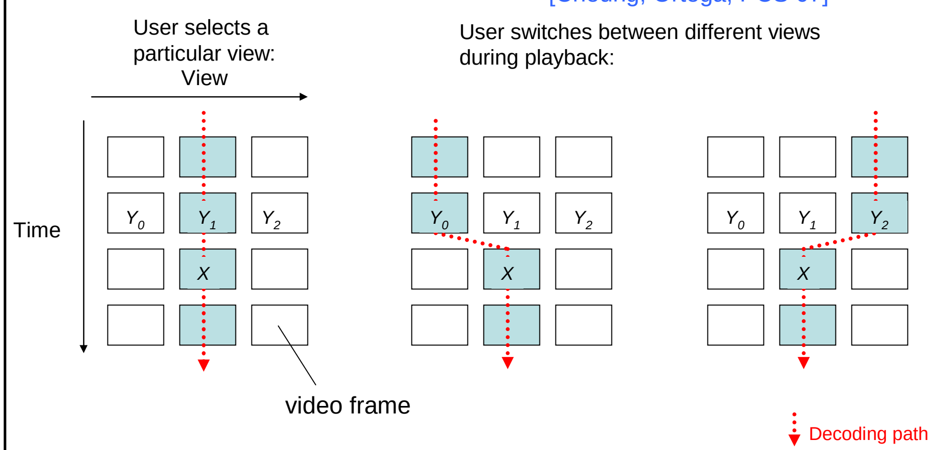
- Free viewpoint switching in multiview video compression
- Forward and backward video playback
- Robust video transmission



Flexible Decoding Example –
Free Viewpoint Switching in Multiview Video

USC Viterbi School of Engineering

[Cheung, Ortega; PCS 07]



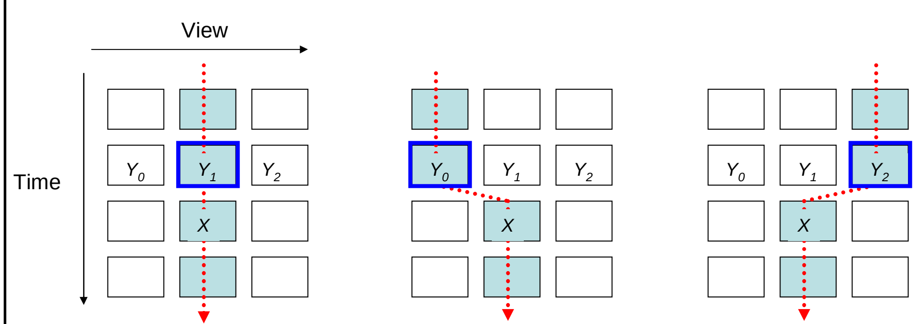
In order to address viewpoint switching, compression schemes have to support **multiple decoding paths**

USC

Free viewpoint switching poses challenges to multiview compression

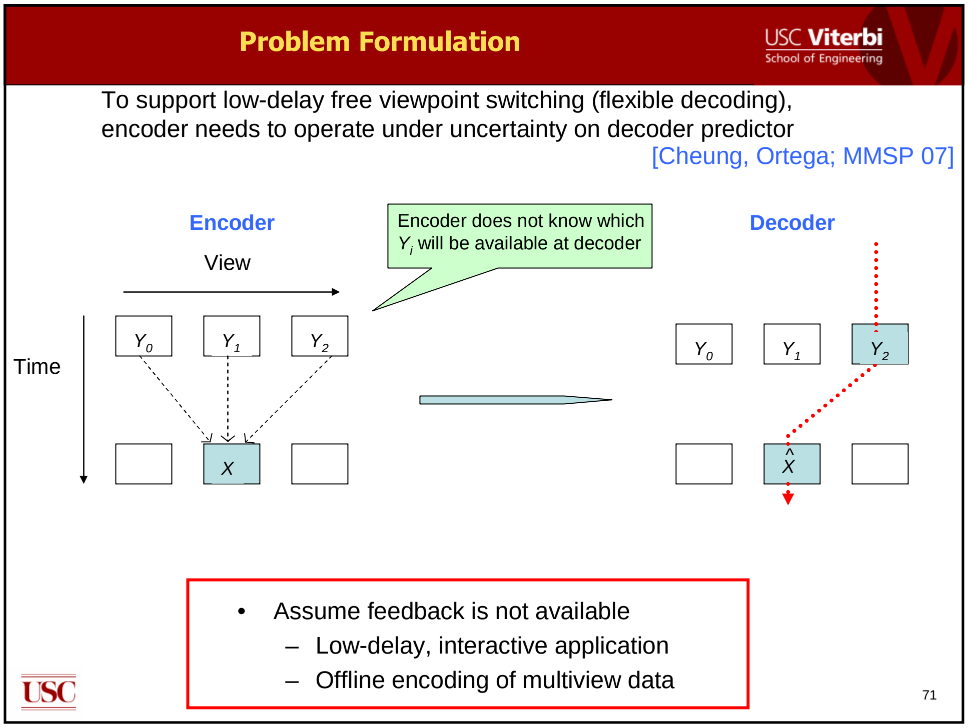
USC Viterbi School of Engineering

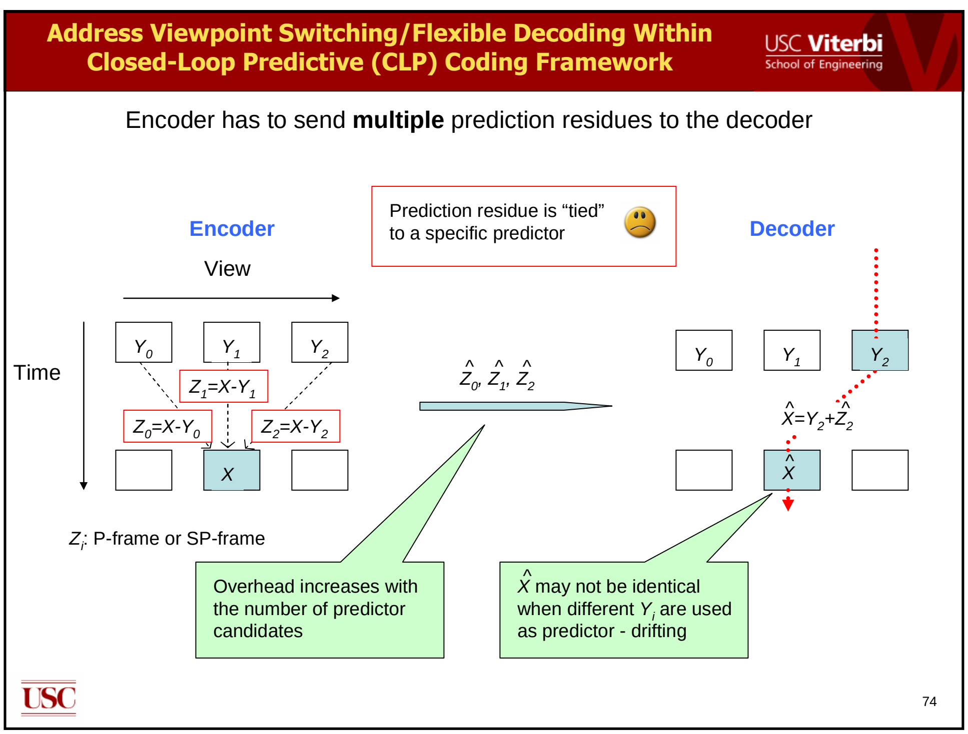
When users can choose among different decoding paths, it is not clear which previous reconstructed frame will be available to use in the decoding



Multiple decoding paths → Either Y_0 or Y_1 or Y_2 will be available at decoder → Uncertainty on predictor status at decoder!!!!

USC


Problem Formulation
USC Viterbi School of Engineering
To support low-delay free viewpoint switching (flexible decoding), encoder needs to operate under uncertainty on decoder predictor
[Cheung, Ortega; MMSP 07]
Encoder
View
Encoder does not know which Y_i will be available at decoder
Decoder
Time
Y_0 Y_1 Y_2
X
Y_0 Y_1 Y_2
$\hat{X}$
• Assume feedback is not available
– Low-delay, interactive application
– Offline encoding of multiview data
USC
71

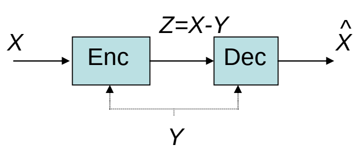

Address Viewpoint Switching/Flexible Decoding Within Closed-Loop Predictive (CLP) Coding Framework
USC Viterbi School of Engineering
Encoder has to send **multiple** prediction residues to the decoder
Prediction residue is "tied" to a specific predictor
Encoder
View
Decoder
Time
Y_0 Y_1 Y_2
$Z_1=X-Y_1$
$Z_0=X-Y_0$
$Z_2=X-Y_2$
X
$\hat{Z}_0, \hat{Z}_1, \hat{Z}_2$
Y_0 Y_1 Y_2
$\hat{X}=Y_2+\hat{Z}_2$
$\hat{X}$
Z_i: P-frame or SP-frame
Overhead increases with the number of predictor candidates
$\hat{X}$ may not be identical when different Y_i are used as predictor - drifting
USC
74

Solution Based on Distributed Source Coding

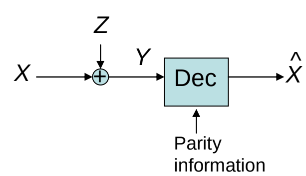
DSC - Virtual Communication Channel Perspective

In DSC, encoder can communicate *X* by sending **parity information**
(E.g., [Girod, Aaron, Rane, Rebollo-Mondero; Proc. IEEE 04])

CLP:



DSC:

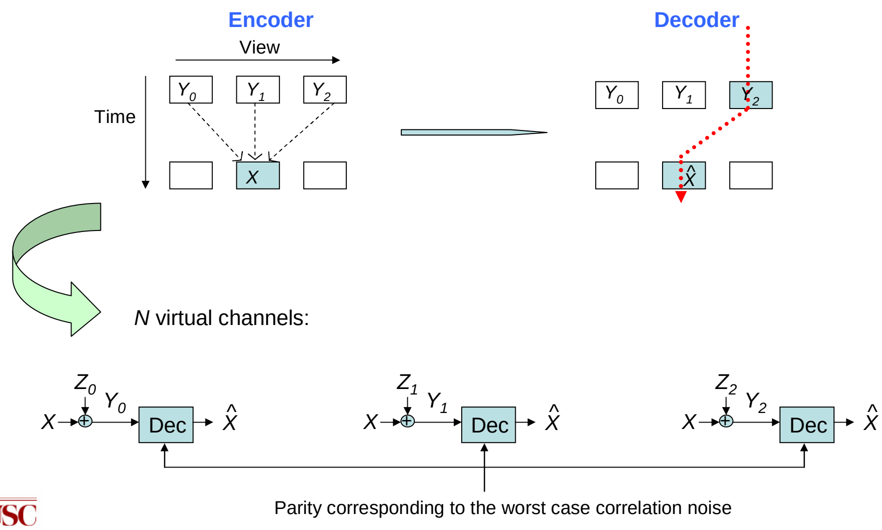


Parity information is **independent** of a specific predictor
- What matters is the **amount** of parity information

Address Viewpoint Switching (Flexible Decoding) Using DSC

USC Viterbi School of Engineering

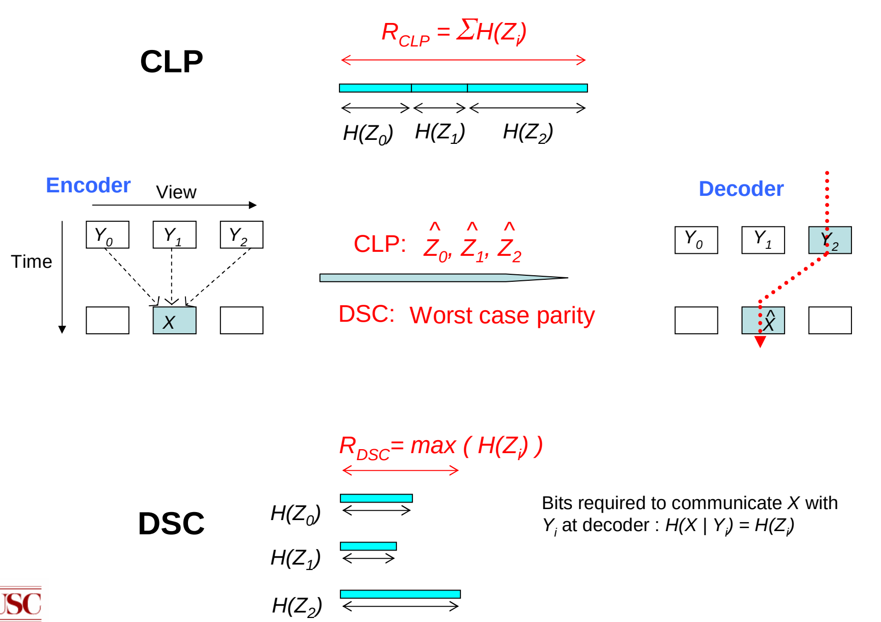
Under predictor uncertainty, encoder can communicate *X* by sending an amount of parity corresponding to the **worst case correlation**



USC 80

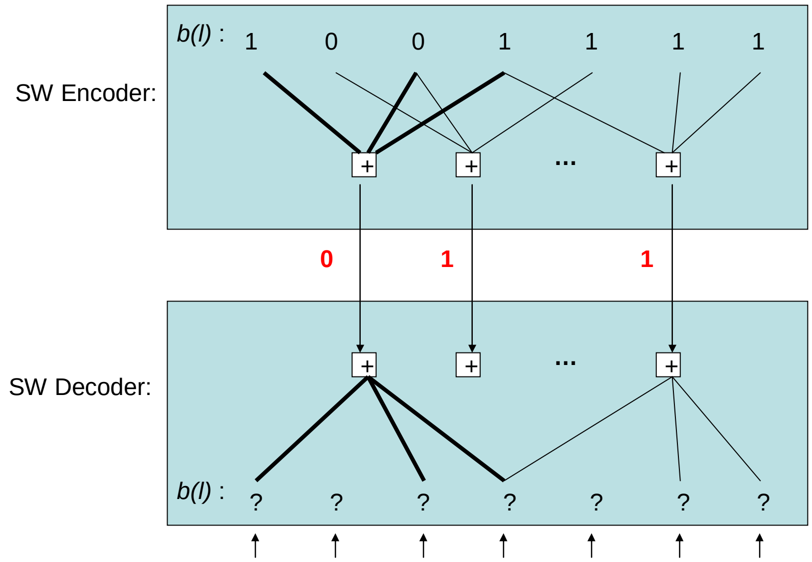
Viewpoint Switching (Flexible Decoding) – CLP vs. DSC

USC Viterbi School of Engineering



USC 81

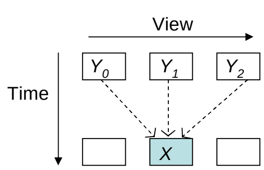
Parity information is independent of a specific predictor

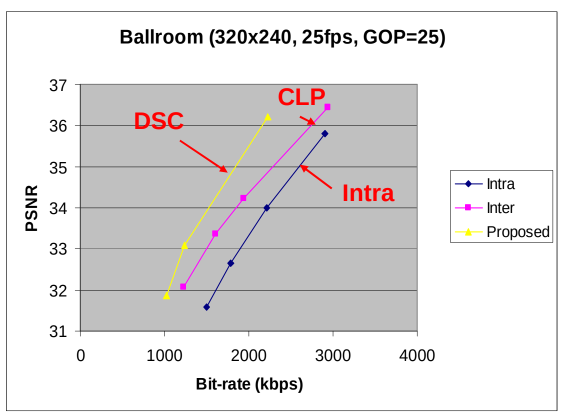


$p(b(l) \mid Y_i, b(l+1), b(l+2), \ldots)$

Experimental Results - Multiview Video Coding

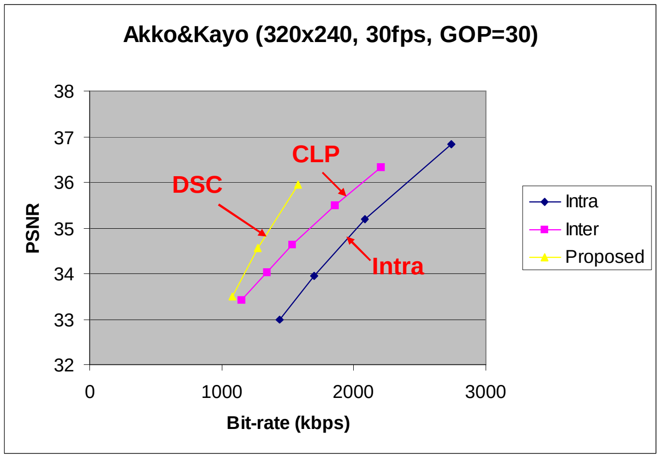
Allow switching from adjacent views: three predictor candidates





Our proposed algorithm out-performs CLP and intra coding

Experimental Results



USC

89

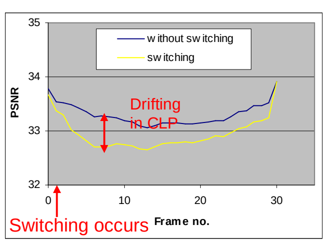
Drifting Experiments

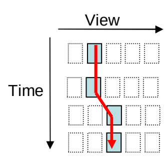
Our proposed algorithm is almost drift-free, since quantized coefficients in DSC coded MB are identically reconstructed

CLP

DSC

Akko&Kayo (GOP=30)





View switching occurs at frame number 2

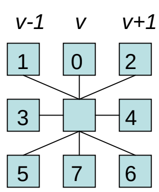
USC

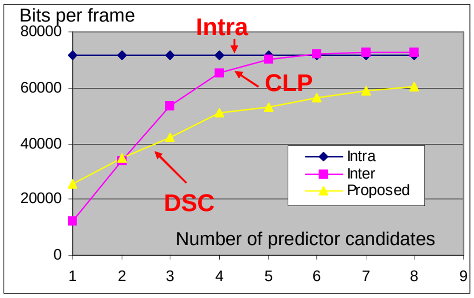
90

Scaling Experiments

USC Viterbi School of Engineering

- Number of coded bits vs. number of predictor candidates





- Bit-rate of DSC-based approach increases at a slower rate compared with CLP
  - An additional candidate incurs more bits only if it has the worst correlation among all candidates

USC

Summary: DSC Application to Viewpoint Switching/Flexible Decoding

USC Viterbi School of Engineering

- DSC-based coding algorithm to address viewpoint switching/flexible decoding
  - **Single** bitstream to support **multiple** decoding paths
  - Parity information **independent** of a specific predictor
  - Overhead depends on the **worst correlation** rather than the number of decoding paths
  - Outperform CLP and intra coding in terms of coding performance
  - Our proposed system is almost drift-free

USC

References

USC Viterbi School of Engineering

- Introduction

  – B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, “Distributed video coding,” Proceedings of the IEEE, Special Issue on Advances in Video Coding and Delivery 93, pp. 71-83, Jan. 2005.
  – Z. Xiong, A. Liveris, and S. Cheng, “Distributed source coding for sensor networks,” IEEE Signal Processing Magazine 21, pp. 80-94, Sept. 2004.

- Recent Advances

  – Guillemot, C., Pereira, F., Torres, L., Ebrahimi, T., Leonardi, R., Ostermann, J., “Distributed Monoview and Multiview Video Coding,” Signal Processing Magazine, IEEE, Volume 24, Issue 5, Sept. 2007, Page(s):67 - 76
  – Workshop on Recent Advances in Distributed Video Coding http://www.discoverdvc.org/Workshop.html

USC