Distributed Source Coding: Foundations, Constructions and Applications

Kannan Ramchandran, UC Berkeley
S. Sandeep Pradhan, UM Ann Arbor
Acknowledgements

- Dinesh Krithivasan (UM)
- Prof. Prakash Ishwar (BU, formerly UCB)
- Prof. David Neuhoff (UM)
- Dr. Vinod Prabhakaran (UIUC, formerly UCB)
- Dr. Rohit Puri (Cisco, formerly UCB)
- June Wang (UCB)
- Chuohao Yeo (UCB)
- ...
- as well as our collaborators on this topic....
Motivation: sensor networks

- Consider correlated nodes $X$, $Y$
- Communication between $X$ and $Y$ expensive.
- Can we exploit correlation without communicating?
- Assume $Y$ is compressed independently. How to compress $X$ close to $H(X|Y)$?
- Key idea: discount $I(X;Y)$. $H(X|Y) = H(X) - I(X;Y)$
Distributed source coding: Slepian-Wolf ’73

Achievable rate-region

Separate encoding of X and Y
Distributed source coding

Source coding with side information: (Slepian-Wolf, ‘73, Wyner-Ziv, ‘76)

- **Lossless coding (S-W):** no loss of performance over when Y is available at both ends if the statistical correlation between X and Y is known.

- **Lossy coding (W-Z):** for Gaussian statistics, no loss of performance over when Y known at both ends.

- **Constructive solutions:** (Pradhan & Ramchandran (DISCUS) DCC ‘99, Garcia-Frias & Zhao Comm. Letters ‘01, Aaron & Girod DCC ’02, Liveris, Xiong & Georghiades DCC ’03,…)

- **Employs statistical instead of deterministic mindset.**
Example: 3-bit illustration

- Let $X$ and $Y$ be length-3 binary data (equally likely), with the correlation: Hamming distance between $X$ and $Y$ is at most 1.
- Example: When $X=[0 \ 1 \ 0]$, $Y$ is equally likely to be $[0 \ 1 \ 0]$, $[0 \ 1 \ 1]$, $[0 \ 0 \ 0]$, $[1 \ 1 \ 0]$.

Encoder
\[ X \rightarrow \text{Encoder} \rightarrow \text{Decoder} \rightarrow \hat{X} = X \]

Decoder

\[ X + Y = \begin{cases} 
0 & 0 & 0 \\
0 & 0 & 1 \\
0 & 1 & 0 \\
1 & 0 & 0 
\end{cases} \]

Need 2 bits to index this.
**Example: 3-bit illustration**

- **X** and **Y** are correlated
- **Y** is available only at decoder (side information)
- **What is the best that one can do?**
  - The answer is still 2 bits!
  - How?

**System 2**

```
Encoder → Decoder
```

\[ X \rightarrow \text{Encoder} \rightarrow \text{Decoder} \rightarrow \hat{X} = X \]

- **Coset-1**
  - \[ \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \]

- **000**
- **001**
- **010**
- **100**
- **111**
- **110**
- **101**
- **011**
Encoder: sends the index of the coset (bin) containing $X$.

Decoder: using index and $Y$, decode $X$ without error.

Coset 1 is a length-3 repetition code

Each coset has a unique associated “syndrome”

Use of syndromes in IT literature: Wyner ’74, Csiszar ‘82
Practical code construction (DISCUS): SP& KR ‘99
Example: geometric illustration

Assume signal and noise are Gaussian, iid
Example: geometric illustration

Assume signal and noise are Gaussian, iid
Example: scalar Wyner-Ziv

- Encoder: send the index of the coset (log23 bits)
- Decoder: decode X based on Y and signaled coset
Outline

- **Session I. Introduction and theory**: 9.00 am-10.00 am
  - Motivation and intuition
  - Distributed source coding foundations
  - Break: 10.00-10.10 am

- **Session II. Constructions**: 10.10 am-10.50 am
  - Structure of distributed source codes
  - Constructions based on trellis codes
  - Constructions based on codes on graphs
  - Break: 10.50-11.00 am

- **Session III. Connections and Applications**: 11.00 am-12.00 noon
  - Overview of connections and applications with snippets
  - Compression of encrypted data
  - Distributed video coding
Outline

- Session I. Introduction and theory : 9.00 am-10.00 am
  - Motivation and intuition
  - Distributed source coding foundations
  - Break: 10.00-10.10 am

- Session II. Constructions: 10.10 am-10.50 am
  - Structure of distributed source codes
  - Constructions based on trellis codes
  - Constructions based on codes on graphs
  - Break: 10.50-11.00 am

- Session III. Connections and Applications: 11.00 am-12.00 noon
  - Overview of connections and applications with snippets
  - Compression of encrypted data
  - Distributed video coding
Source coding: lossless case [Shannon ’49]

- Source alphabet $X$
- Source distribution $\sim p_X(x)$

- Encoder: $e: X^N \rightarrow \{1, 2, \ldots, 2^{NR}\}$
- Decoder: $f: \{1, 2, \ldots, 2^{NR}\} \rightarrow X^N$

- Goal: minimize rate $R$ such that probability of decoding error $\sim 0$
- Answer: $R \geq H(X)$

- Idea: index only typical sequences

\[ \text{Set of all N-length sequences} \]
\[ (\text{Size} \approx 2^{N \log|X|}) \]
Source coding: lossless case

- Source alphabet $X$
- Source distribution $p_X(x)$

- Encoder: $e: X^N \rightarrow \{1, 2, \ldots, 2^{NR}\}$
- Decoder: $f: \{1, 2, \ldots, 2^{NR}\} \rightarrow X^N$

- Goal: minimize rate $R$ such that probability of decoding error $\sim 0$
- Answer: $R \geq H(X)$

- Idea: index only typical sequences
  - Probability of typical set $\sim 1$

Set of all $N$-length sequences
(Size $\approx 2^{N \log |x|}$)

Set of typical sequences
(Size $\approx 2^{NH(X)}$)
Source coding: lossy case [Shannon ’58]

- Distortion function: $d(x, \hat{x})$

- Goal: minimize rate $R$ such that expected distortion $< D$

- Answer: $R \geq R(D) = \min_{p(\hat{x}|x): Ed(X, \hat{X}) \leq D} I(X; \hat{X})$

- Idea
  - Cover typical set with “spheres” of radius $D$,
  - Index these “spheres”
  - Size of each “sphere” $\approx 2^{NH(X|\hat{X})}$
  - Rate $= H(X) - H(X | \hat{X}) = I(X; \hat{X})$
  - Sequences which get the same index are nearby
Source coding: lossy case [Shannon ’58]

- Distortion function: \( d(x, \hat{x}) \)
- Goal: minimize rate \( R \) such that expected distortion \( < D \)
- Answer: \( R \geq R(D) = \min_{p(\hat{x}|x): Ed(X, \hat{X}) \leq D} I(X; \hat{X}) \)
- Idea
  - Cover typical set with “spheres” of radius \( D \),
  - Index these “spheres”
  - Size of each “sphere” \( \approx 2^{NH(X|\hat{X})} \)
  - Rate = \( H(X) - H(X | \hat{X}) = I(X; \hat{X}) \)
  - Sequences which get the same index are nearby
**Source coding: lossy case [Shannon ’58]**

- **Distortion function:** $d(x, \hat{x})$
  
- **Goal:** minimize rate $R$ such that expected distortion < $D$
  
- **Answer:** $R \geq R(D) = \min_{p(\hat{x}|x):Ed(X,\hat{X})\leq D} I(X; \hat{X})$
  
- **Idea**
  - Cover typical set with “spheres” of radius $D$,
  - Index these “spheres”
  - Size of each “sphere” $\approx 2^{NH(X|\hat{x})}$
  - Rate $= H(X) - H(X | \hat{X}) = I(X; \hat{X})$
  - Sequences which get the same index are nearby
Source coding w/ side information: lossless case

- Source $X$
- Side information $Y$
- Goal: minimize rate $R$ s.t. prob. of reconstruction error $\sim 0$
- Answer: $R = H(X|Y)$
- Idea
  - Given side information sequence $Y^N$, index conditionally typical sequences of $X^N$ given $Y^N$

[Gray ’73, Berger ’71]
Source coding w/ side information: lossless case

- Source $X$
- Side information $Y$
- Goal: minimize rate $R$ s.t. prob. of reconstruction error $\sim 0$
- Answer: $R = H(X|Y)$

Idea
- Given side information sequence $Y^N$, index conditionally typical sequences of $X^N$ given $Y^N$
Source coding w/ side information: lossless case

- Source $X$
- Side information $Y$
- Goal: minimize rate $R$ s.t. prob. of reconstruction error $\approx 0$
- Answer: $R = H(X|Y)$

Idea
- Given side information sequence $Y^N$, index conditionally typical sequences of $X^N$ given $Y^N$

Conditionally typical set of size $\approx 2^{NH(X|Y)}$
Source coding w/ side information: lossy case

- Source $X$
- Side information $Y$
- Source distribution $\sim p_{X|Y}(x|y)$
- Goal: minimize rate $R$ such that expected distortion $< D$
Source coding w/ side information: lossy case

- Source $X$
- Side information $Y$
- Source distribution $\sim p_{X|Y}(x|y)$

Goal: minimize rate $R$ such that expected distortion $< D$

Answer: conditional rate-distortion function

$$R \geq R_{X|Y}(D) = \min_{p(\hat{x}|x,y): Ed(X, \hat{X}) \leq D} I(X; \hat{X} | Y)$$

Idea
- Given side information sequence $Y^N$, cover the conditionally typical set of $X^N$ given $Y^N$ using “spheres” of radius $D$
- Index these spheres
Source coding w/ side information: lossy case

- Source $X$
- Side information $Y$
- Source distribution $\sim p_{X|Y}(x|y)$
- Goal: minimize rate $R$ such that expected distortion $< D$
- Answer: conditional rate-distortion function
  \[
  R \geq R_{X|Y}(D) = \min_{p(\hat{x}|x,y): Ed(X,\hat{X}) \leq D} I(X;\hat{X} | Y)
  \]
- Idea
  - Given side information sequence $Y^N$, cover the conditionally typical set of $X^N$ given $Y^N$ using “spheres” of radius $D$
  - Index these spheres

Decoder
Encoder

Conditionally typical $X$-sequences

Y-sequence

[Gray ’73, Berger ’71]
Source coding w/ side information: lossy case

- Source $X$
- Side information $Y$
- Source distribution $\sim p_{X|Y}(x|y)$
- Goal: minimize rate $R$ such that expected distortion $< D$
- Answer: conditional rate-distortion function
  \[ R \geq R_{X|Y}(D) = \min_{p(\hat{x}|x,y): Ed(X,\hat{X}) \leq D} I(X;\hat{X} | Y) \]
- Idea
  - Given side information sequence $Y^N$, cover the conditionally typical set of $X^N$ given $Y^N$ using “spheres” of radius $D$
  - Index these spheres

[Gray ’73, Berger ’71]
**Source coding w/ SI at decoder only: lossless**

- **Source** $X$
- **Side information** $Y$
- **Source distribution** $\sim p_{X|Y}(x|y)$
- **Goal**: minimize rate $R$ s.t. prob. of reconstruction error $\sim 0$

**Idea**
- Typical $X$-sequences which are far apart given the same index
- Induces a partition on the space of $X$ : binning
- Any valid $Y$-sequence $\rightarrow$ there do not exist more than one conditionally typical $X$-sequence having the same index

[Slepian-Wolf ’73]
Source coding w/ SI at decoder only: lossless

- **Source** $X$
- **Side information** $Y$
- **Source distribution** $\sim p_{X|Y}(x|y)$
- **Goal**: minimize rate $R$ s.t. prob. of reconstruction error $\sim 0$

**Idea**
- Typical $X$-sequences which are far apart given the same index
- Induces a partition on the space of $X$ : binning
- Any valid $Y$-sequence $\rightarrow$ there do not exist more than one conditionally typical $X$-sequence having the same index
Source coding w/ SI at decoder only: lossless

- Source $X$
- Side information $Y$
- Source distribution $\sim p_{X|Y}(x|y)$
- Goal: minimize rate $R$ s.t. prob. of reconstruction error $\sim 0$
- Idea
  - Typical $X$-sequences which are far apart given the same index
  - Induces a partition on the space of $X$: binning
  - Any valid $Y$-sequence $\rightarrow$ there do not exist more than one conditionally typical $X$-sequence having the same index

[Source coding w/ SI at decoder only: lossless: Slepian-Wolf '73]

Encoder $X$ → Decoder $X$

Conditionally typical set of $X|Y$
Source coding w/ SI at decoder only: lossless

- Source $X$
- Side information $Y$
- Source distribution $\sim p_{X|Y}(x|y)$
- Goal: minimize rate $R$ s.t. prob. of reconstruction error $\sim 0$

Idea
- Typical $X$-sequences which are far apart given the same index
- Induces a partition on the space of $X$: binning
- Any valid $Y$-sequence $\rightarrow$ there do not exist more than one conditionally typical $X$-sequence having the same index

[Source: Slepian-Wolf '73]

Conditionally typical set of $X|Y$
Source coding w/ SI at decoder only: lossless

- Set of $X$-sequences that get the same index ↔ channel code for the fictitious channel with input $X$, output $Y$

- Channel: input distribution $p_X(x)$, transition probability $\sim p_{X|Y}(x|y)$

- Max. reliable rate of transmission $= I(X;Y)$

- This rate comes for free from this fictitious channel

  $$ R = H(X) - I(X;Y) = H(X) - [H(X) - H(X|Y)] = H(X|Y) $$

- Source space partitioned into cosets (shift) of channel codebooks

- No loss in performance for lack of $Y$ at encoder

[Slepián-Wolf ’73]
Source coding w/ SI at decoder only: lossy

- Source $X$
- Side information $Y$
- Distortion function $d(x, \hat{x})$
- Goal: minimize rate $R$ such that expected distortion $< D$

[Wyner Ziv ’76]
Quantize $X$ to some intermediate reconstruction $U$

From standard R-D theory, this would incur a rate of $I(X; U)$

Apply source coding with SI idea losslessly

New fictitious channel has input $U$, and output $Y$

This gives a rebate in rate of $I(U; Y)$

Total rate $= I(X; U) - I(U; Y)$
Source coding w/ SI at decoder only: lossy

- Quantize $X$ to some intermediate reconstruction $U$
- From standard R-D theory, this would incur a rate of $I(X; U)$
- Apply source coding with SI idea losslessly
- New fictitious channel has input $U$, and output $Y$
- This gives a rebate in rate of $I(U; Y)$
- Total rate $= I(X; U) - I(U; Y)$
Source coding w/ SI at decoder only: lossy

- Encoder does not observe $Y$

- Choosing $p(u|x)$ fixes the joint distribution of $X,Y,U$ using Markov chain condition $Y \rightarrow X \rightarrow U$ as $p(y)p(x|y)p(u|x)$

- The decoder has two looks at $X$: through $U$, through $Y$

- Get an optimal estimate of $X$ given $U$ and $Y$: $\hat{X} = g(U,Y)$

- SI $Y$ is used twice: recovering $U$, estimating $X$

- $R \geq R_{WZ}(D) = \min_{p(u|x): Ed(X,\hat{X}) \leq D} I(X;U) - I(U;Y)$

Conditionally typical sequences of $U|Y$ Y-sequence

[Wyner Ziv ’76]
Remark

- Quantizer for the source $X$ is partitioned into cosets (shift) of channel codebooks for the fictitious channel with i/p $U$ and o/p $Y$

- Contrast between two kinds of many-to-one encoding functions:
  - Quantization: sequences that get the same index are nearby
  - Binning: sequences that get the same index are far apart
Example: Gaussian with quadratic distortion

Lossy source coding with no side information

- X is zero-mean Gaussian with variance $\sigma_x^2$

- Quadratic distortion: $d(x, \hat{x}) = (x - \hat{x})^2$

- $R(D) = \frac{1}{2} \log \left( \frac{\sigma_x^2}{D} \right)$

- Test channel is given by:

  $X \xrightarrow{\mu} \mu + q \xrightarrow{\hat{X}}$

  \[ \Rightarrow p(\hat{x} \mid x) = \text{Gaussian} \]

  \[
  \text{mean} = \mu_x \\
  \text{var} = \frac{D(\sigma_x^2 - D)}{\sigma_x^2} \\
  \mu = \frac{\sigma_x^2 - D}{\sigma_x^2}
  \]
Example: Gaussian with quadratic distortion

Lossy source coding with side information

- $X = Y + N$, where $N$ is zero-mean Gaussian with variance $\sigma_N^2$
- $Y$ is arbitrary and independent of $N$
- Quadratic distortion: $d(x, \hat{x}) = (x - \hat{x})^2$
- $R_{X|Y}(D) = \frac{1}{2} \log \left( \frac{\sigma_n^2}{D} \right)$
- Test channel is given by:

$$p(\hat{x} \mid x, y) = \text{Gaussian}$$

mean $= \alpha x + (1 - \alpha) y$

$\text{var} = \frac{D(\sigma_n^2 - D)}{\sigma_n^2}$

$\alpha = \frac{\sigma_n^2 - D}{\sigma_n^2}$
Example: Gaussian with quadratic distortion

Lossy source coding with side information at decoder only

- Source, side information, and distortion as before
- \( R_{WZ}(D) = \frac{1}{2} \log \left( \frac{\sigma_n^2}{D} \right) \) → no performance loss for lack of \( Y \) at encoder
- Test channel when SI is present at both ends
- Test channel when SI is present at decoder only

\[
\Rightarrow p(u \mid x) = \text{Gaussian} \\
\text{mean} = \alpha x \\
\text{var} = \frac{D(\sigma_n^2 - D)}{\sigma_n^2} \\
\hat{X} = U + (1 - \alpha)Y
\]
Distributed source coding: lossless case

- Minimize rate pair $R_X, R_Y$ such that probability of decoding error $\sim 0$

![Diagram of distributed source coding](image)

- Achievable rate-region:
  - Separate encoding of $X$ and $Y$
Example

- X and Y → length-7 equally likely binary data with Hamming distance between them at most 1.
  - $H(X) = 7$ bits
  - $H(Y|X) = 3$ bits = $H(Y|X)$
  - $H(X,Y) = 10$ bits

**Achievable Rate-Region**

*Separate encoding of X and Y*
Distributed source coding: lossy case

- Minimize rate pair $R_X, R_Y$ such that
  
  $$E[d_X(X, \hat{X})] \leq D_X, \text{ and } E[d_Y(Y, \hat{Y})] \leq D_Y$$

- Optimal performance limit: open problem!

- Approach: [Berger-Tung ’77]
  - Quantize $Y$ to $V$
  - Treat $V$ as side information
Distributed source coding: lossy case

- Berger-Tung achievable rate region

\[ R_X \geq I(X;U) - I(U;V) \]
\[ R_Y \geq I(Y;V) - I(U;V) \]
\[ R_X + R_Y \geq I(X;U) + I(Y;V) - I(U;V) \]

- For every choice of \( p_{U|X}(u|x) \), \( p_{V|Y}(v|y) \) that satisfies distortion constraints

- Overall rate region is the union of such regions

- Can be easily generalized to more general distortion functions

- Shown to be tight in some special cases
Remarks

- All results → random quantization and binning

- Structured random codes may give a better performance than unstructured random codes [Korner-Marton ’79]

- Structured codes for quantization and binning is a topic of active research.
BREAK
10.00 AM-10.10 AM
Outline

- Session I. Introduction and theory: 9.00 am-10.00 am
  - Motivation and intuition
  - Distributed source coding foundations
  - Break: 10.00-10.10 am

- Session II. Constructions: 10.10 am-10.50 am
  - Structure of distributed source codes
  - Constructions based on trellis codes & codes on graphs
  - Break: 10.50-11.00 am

- Session III. Connections and Applications: 11.00 am-12.00 noon
  - Overview of connections and applications with snippets
  - Compression of encrypted data
  - Distributed video coding
Active area of recent research

- **Theory**
  - D. Slepian & J. Wolf ('73)
  - A. Wyner ('74)
  - A. Wyner & J. Ziv ('76)
  - I. Csiszar ('82)
  - Zamir et al ('02, '03, '04)
  - S. Pradhan & K. Ramchandran ('05)
  - Many More...

- **Source coding with side information – I.I.D. Sources**
  - A. Orlitsky ('93, Graph Theoretic)
  - S. Pradhan & K. Ramchandran ('99)
  - Y. Zhao & J. Garcia-Frias ('02, larger alphabets)
  - A. Liveris, Z. Xiong, & C. Georgiades ('02)
  - D. Schonberg, S. Pradhan, & K. Ramchandran ('02)
  - P. Mitran & J. Bajcsy ('02)
  - A. Aaron & B. Girod ('02)
  - A. Liveris, Z. Xiong, & C. Georgiades ('03)
  - J. Li, Z. Tu, & R. Blum ('04)
  - M. Sartipi & F. Fekri ('05)

- **Source coding with side information – Correlated Sources**
  - J. Garcia-Frias & W. Zhong ('03)
  - D. Varodayan, A. Aaron, & B. Girod ('06)
Example

- X and Y -> length-7 equally likely binary data with Hamming distance between them at most 1.
  - H(X)= ? bits
  - H(Y|X)= ? bits = H(Y|X)
  - H(X,Y)=? bits

Answer:
• H(x)=H(Y)=7 bits, H(X,Y)=10 bits
• Use (7,4,3) Hamming code
• Send Y as is (7 bits)
• Send syndrome for X (3 bits)
Symmetric Coding

- **Example:**
  - X and Y -> length-7 equally likely binary data.
  - Hamming distance between X and Y is at most 1

- **Solution 1:**
  - Y sends its data with 7 bits.
  - X sends syndromes with 3 bits.
  - \{ (7,4) Hamming code \} -> Total of 10 bits

- **Solution 2:** source splitting [Willems ‘88, Urbanke-Rimoldi ‘97]

- Can correct decoding be done if X and Y send 5 bits each?
Symmetric Coding

- Solution: Map valid \((X,Y)\) pairs into a coset matrix [SP & KR ‘00]

### Coset Matrix

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>...</th>
<th>31</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Construct 2 codes, assign them to encoders
- Encoders → send index of coset of codes containing the outcome
Symmetric Coding

\[ G = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \Rightarrow \quad G_1 = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \end{pmatrix} \quad G_2 = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \]

\[ G_1, G_2 \Rightarrow 2 \times 7 \Rightarrow \text{ Syndromes are 5 bits long} \]

- Decoder: Find a pair of codewords (one from each coset) that satisfy the distance criterion
- There exists a fast algorithm for this
- This concept can be generalized to Euclidean-space codes.
The rate region is:

\[ \begin{align*}
R_x, R_y : & \quad R_x \geq 3, R_y \geq 3 \\
& \quad R_x + R_y \geq 10
\end{align*} \]

- All 5 optimal points can be constructively achieved with the same complexity.
- All are based on a single linear code.
- Can be generalized to arbitrary statistics [Schonberg et al. 2002]
LDPC Codes: Brief Overview

- Need linear codes ⇒ use LDPC codes.

- Class of capacity approaching linear block codes.

- Sparse parity check matrix depicted by Tanner graph
  - Circles represent bits.
  - Squares represent constraints.
**LDPC Codes Overview: decoding**

- Decoded via message passing algorithm.
- Messages passed in two phases.
  - Update rules:
    
    \[
    \begin{align*}
    v_{is}(x_i) &= \prod_{t \in N(i) \setminus s} \mu_{ti}(x_i) \\
    \mu_{si}(x_i) &= \sum_{x_{N(s) \setminus i}} \left( f_s(x_{N(s)}) \prod_{j \in N(s) \setminus i} v_{js}(x_j) \right)
    \end{align*}
    \]

- Distribution of each variable estimated after \( n \) iterations.

    \[
    p(x_i) = \frac{1}{Z} \prod_{s \in N(i)} \mu_{si}(x_i)
    \]
Source coding w/ side information at decoder

- $X = Y + N$, $Y$ is arbitrary
- $N$ is zero-mean Gaussian with variance $\sigma_n^2$
- $Y$ and $N$ are independent
- Quadratic distortion: $d(x, \hat{x})$
- Performance limit: $R_{WZ}(D) = \frac{1}{2} \log \left( \frac{\sigma_n^2}{D} \right)$
- Key idea: source codebook partitioned into cosets of channel codebooks
- Goal: computationally efficient way to construct
  - Source codebook (quantizer) with an encoding procedure
  - Partition of the quantizer into cosets of channel codebooks
Symmetric Coding: illustration

- Source bits, compressed bits, and LDPC code applied to Y

- Source bits, compressed bits, and LDPC code applied to K

- Correlation constraints
<table>
<thead>
<tr>
<th>Standard source coding</th>
<th>Distributed source coding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boundary gain ~ 6 - 9 dB: achieved by entropy coding</td>
<td>Coding gain ~ 6 - 9 dB: achieved by partition using LDPC channel codes</td>
</tr>
<tr>
<td>Granular gain ~ 1.53 dB: achieved by vector quantization</td>
<td>Granular gain ~ 1.53 dB: achieved by vector quantization</td>
</tr>
</tbody>
</table>

- **Bounding region**
- **Voronoi region**
<table>
<thead>
<tr>
<th><strong>Standard source coding</strong></th>
<th><strong>Distributed source coding</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>▪ VQ and entropy coding can be done independently</td>
<td>▪ VQ and partition using channel codes cannot be done independently</td>
</tr>
<tr>
<td>▪ TCQ gets 1.36 dB of granular gain =&gt; within 0.15 dB from R-D limit</td>
<td>▪ Algebraic structure of TCQ does not “gel” well with that of LDPC codes</td>
</tr>
<tr>
<td></td>
<td>▪ Need new block-coded quantization techniques!</td>
</tr>
</tbody>
</table>
Role of “source codes”


Joint quantization and estimation:

Active source codeword: Codeword -> X is quantized.

Quantization:  
- Quantize X to W
- W => “Source Codes”

Estimation: Estimate X using W and Y.
Role of “channel codes”

- “Channel Codes”: Reduce source coding rate by exploiting correlation
- Partition of the “source codes” into cosets of “channel codes”:
  
  ![Diagram](image)

  - Source Codewords (elements of the set $W$)
  - A subset of $W$ -> channel coset code -> Channel $p(y|w)$.

- Y and $W$ are correlated => induces an equivalent channel $p(y|w)$.
- Build “channel coset codes” on $W$ for channel $p(y|w)$.
Role of “channel codes”

- Partition $W$ into cosets of such “channel codes”.

Decoder:
- Recovers active source codeword by channel decoding $Y$ in given coset
- Channel decoding fails => Outage
Encoder and Decoder Structure

Encoder

1. Find index of active codeword

2. Compute index of the coset containing the active codeword

Decoder

1. Channel decode $Y$ in the coset $U$ to find the active codeword

2. Estimate $X$
Distributed Source Coding Theory

- Source coding theory
  - Quantization
  - Indexing
  - Fidelity criterion

- Channel coding theory
  - Algebraic structure
  - Minimum distance
  - Prob. of decoding error

- Estimation theory
  - Estimation with rate constraints

Intricate Interplay
**Basic Concept**

**Illustrative Example:**

Consider a fixed-length scalar quantizer (say with 8 levels)

Partition:

4 Cosets
Trellis based coset construction

Example: Rate of transmission= 1 bit/source sample. Quantizer: fixed-length scalar quantizer ->8 levels.

- $C = \{r_0, r_1, r_2, r_3, r_4, r_5, r_6, r_7\} \Rightarrow$ set of codewords.

- $C^N$ has $2^{3N}$ sequences

- Partition $C^N$ into $2^N$ cosets each containing $2^{2N}$ sequences.

- Use Ungerboeck trellis for such partitioning.

Trellis Coding:

- Sequences generated by this machine form a coset in space $C^L$

- Coset $\Rightarrow 2^{2N}$ sequences.
Trellis Partitioning:

- Source Codebook $= C^N$
- Channel Codebook= set of sequences generated by the finite state machine
- Task: partition $C^N$ into $2^N$ cosets, containing $2^{2N}$ sequences (in a computationally efficient way)

- Fast Encoding: Send syndrome sequence of active codeword
- Fast Decoding: Modified Viterbi algorithm using relabeling.
Trellis Partitioning

- Connection with earlier picture
  \( C = \{r_0, r_1, \ldots, r_7\} \rightarrow \text{codewords of scalar quantizer} \)

\[ Q : \{0,1\}^3 \rightarrow C \]

Set of N-length sequences generated by the finite state machine
Simulation Results

- **Model:**
  - Source: \( X \sim \text{i.i.d. Gaussian} \)
  - Side information: \( Y = X + N \), where \( N \sim \text{i.i.d. Gaussian} \)
  - Correlation SNR: ratio of variances of \( X \) and \( N \)
  - Normalized distortion: ratio of distortion and variance of \( X \)
  - Effective source coding rate = 1 bit per source sample

- **Quantizers:**
  - Fixed-length scalar quantizers with 4, 8 and 16 levels

- Shannon R-D Bound: distortion= -6.021 dB at 1 bit/sample.
Simulation Results

- **Distortion Performance**
  - 8-levels
  - 4-levels
  - 16-levels
  - Wyner-Ziv Bound
    - Rate = 1 b/s
  - Wyner-Ziv Bound
    - Rate = 2 b/s

- **Probability of error**
  - (uncoded system)

- **Graphs**:
  - Left: Correlation-SNR in dB vs. Normalized Distortion in dB
  - Right: Correlation-SNR in dB vs. Probability of error (in exponents of 10)
Probability of Error:

4-level root scalar quantizer

Trellis coset coded system

8-level root scalar quantizer

Gains (at C-SNR=18 dB) : Theoretical = 18 dB.
over Shannon bound      DISCUS = 14 dB  at Prob. of error ≤ 10^{-4}
Approaches based on codes on graph

- Trellis codes → codes on graph to effect this partition

- Need good source code and good channel code

- Start with simple (not so good) source codebook and very good channel codebooks.

- Use belief propagation at the decoder to recover active source codeword
Reminder: graphical models

- Factor Graphs
  - Circles: Variables, Squares: Constraints

- Graphical representation for linear transformation
  - $Y$ – source bits, $U$ – compressed bits
  - Squares – Linear transformation Equations

- Transformation inversion: Belief propagation
  - Iterative application of inference algorithm
Approaches based on codes on graph

Xiong et al., Garcia-Frias et al.

Example: Rate of transmission = 1 bit/source sample.
Quantizer: fixed-length scalar quantizer -> 8 levels.

- \( C = \{ r_0, r_1, r_2, r_3, r_4, r_5, r_6, r_7 \} \rightarrow \text{set of codewords.} \)
- \( C^N \) has \( 2^{3N} \) sequences
- Partition \( C^N \) into \( 2^N \) cosets each containing \( 2^{2N} \) sequences.

Multi-level Coding using binary block codes of code rate 2/3

- Sequences generated by this machine form a coset in space \( C^L \)
- Coset -> \( 2^{2N} \) sequences.
- within 1.53 dB from R-D limit
Partitioning based on LDPC codes

Connection with earlier picture:

\[ C = \{ r_0, r_1, r_2, r_3, r_4, r_5, r_6, r_7 \} \Rightarrow \text{codewords of scalar quantizer} \]

\[ Q : \{0,1\}^3 \rightarrow C \]

Set of N-length sequences generated by the block code
Binary Memoryless Sources

- X, Y: binary symmetric correlated sources

- Correlation: \( Y = X \oplus Z \), Z is Bernoulli(p) and independent of X

- Distortion: \( d(x, \hat{x}) = w_H(x \oplus \hat{x}) \)

- Goal:
  - Build a quantizer for X (U represents the quantized version)
  - Build a channel code for the channel with i/p U and o/p Y
  - Put a linear structure on both quantizer and channel code
  - Channel code is a subcode of the quantizer \( \Rightarrow \) induces a coset partition
Binary Memoryless Sources

- Linear codes:

- Channel code:
  - Theory of binary linear channel codes $\rightarrow$ well-developed
  - LDPC codes with belief propagation (BP) algorithm
  - Gets the ultimate rebate $I(U;Y)$

- Block quantizer:
  - LDPC codes are not good quantizers, BP fails for quantization
  - A new theory of binary block quantizers
  - LDGM (low-density generator matrix) codes
Belief propagation vs Survey propagation

Channel decoding: belief propagation approximates min. distance decoding

Quantization: survey propagation approximates min. distance encoding
Lattice codes:

Fine hexagonal lattice = source quantizer
Coarse hexagonal lattice = channel codebook
BREAK
10.50 AM-11.00 AM
Outline

- Session I. Introduction and theory: 9.00 am-10.00 am
  - Motivation and intuition
  - Distributed source coding foundations
  - Break: 10.00-10.10 am

- Session II. Constructions: 10.10 am-10.50 am
  - Structure of distributed source codes
  - Constructions based on trellis codes & codes on graphs
  - Break: 10.50-11.00 am

- Session III. Connections and Applications: 11.00 am-12.00 noon
  - Overview of connections and applications with snippets
  - Compression of encrypted data
  - Distributed video coding
Connections and Applications

- Fundamental duality between source coding and channel coding with side-information
  - Media security: data-hiding, watermarking, steganography
- Digital upgrade of legacy analog systems
- M-channel Multiple Description codes
- Robust rate-constrained distributed estimation (CEO problem)
- Media broadcast using hybrid analog/digital techniques

- Distributed compression in sensor networks
- Compression of encrypted data
- Distributed Video Coding
Duality b/w source & channel coding with SI

- Source coding with side information (SCSI)

  \[ X \xrightarrow{m} \text{Encoder} \rightarrow Y \xrightarrow{\hat{m}} \text{Decoder} \]

  Sensor networks, video-over-wireless, multiple description, secure compression

- Channel coding with side information (CCSI)

  \[ m \xrightarrow{X} \text{Encoder} \rightarrow S \]

Watermarking, audio data hiding, interference pre-cancellation, multi-antenna wireless broadcast.

SP, J. Chou and KR, Trans. on IT, May 2003
The encoder sends watermarked image $X$.
Attacker distorts $X$ to $Y$.
Decoder extracts watermark from $Y$.

Embed (authentication) signature that is robust.
Application: digital audio/video simulcast

Ex.: Upgrading NTSC to HDTV with a digital side-channel. (also PAL+)

Digital Encoder

Noise

Decoder

Digital
Application: spectrum “recycling”

What is the optimal tradeoff between simultaneously delivered analog and digital quality? ➔ Need a combination of SCSI/CCSI

Hybrid Analog-Digital Simulcast

What is noise for analog receiver is music for digital receiver!

Multiple Description (MD) coding

- Packet erasure model: *some* subset of packets reach decoder.
- Connection to DSC: Uncertainty re. *which* packets reach decoder?
- Fundamental connection between MD coding and distributed source coding: leads to **new achievable rate results**!

*R.Puri, SP & KR (Trans. on IT- Jan 04, Apr 05)*
Sensors make noisy (correlated) measurements of a physical quantity $X$, e.g.,
temperature, pressure, seismic waves, audio data, video signals, etc.

Central decoder needs to estimate $X$ (many-one topology).

Power conservation, node failures, communication failure $\Rightarrow$ need **robustness**.
Robust distributed rate-constrained estimation

- For a \((k, k)\) reliable Gaussian network, full range of rate-MSE tradeoff: [Oohama, IT 1998].
Robust distributed rate-constrained estimation

- For a $(k, k)$ **reliable Gaussian** network, full range of rate-MSE tradeoff: [Oohama, IT 1998, Prabhakaran, Tse & KR ISIT 2001].

- For an $(n, k)$ **unreliable** Gaussian network, can match above performance for the reception of **any** $k$ packets and get better quality upon receiving more packets!
  - $\Rightarrow$ Robustness without loss in performance.

```
\begin{align*}
  m \geq k & \text{ links are up} \\
  O(1/m) & \\
  0 \quad R \ldots kR \quad mR \quad nR \quad \text{active-network rate}
\end{align*}
```

*P. Ishwar, R. Puri, SP & KR, IPSN’03.*
Adaptive filtering for distributed compression in sensor networks

Deployment setup

- Network consists of many sensors, a few controllers, and a few actuators.
- Sensors give their measurements to controllers, which process them and make decisions.
- Many sensors have highly correlated data.
- It would be beneficial to exploit this correlation to compress sensor readings.
Challenges of Real World

- Theory says what is possible given the correlation.
- Codes exist which achieve bounds when correlation is known.
- How does one find the correlation?
Setup

1. Controller receives uncoded data from sensors
2. Breaks them up into clusters s.t. nodes within cluster are highly correlated
3. Tells each cluster what code-book to use
Tree-Structured Code

- Depth of tree specifies number of bits used for encoding

Path in the tree specifies the encoded value.

- Can tolerate $2^{i-1}\Delta$ of correlation noise using an $i^{th}$ level codebook
How Much Compression?

- Sensor nodes measure $X$, data controller node has $Y$
- Controller needs to estimate number of bits, $i$, it needs from sensor nodes for $X$.

- $X = Y + N$; $N = \text{correlation noise}$

- $P[|N| > 2^{i-1} \Delta] \leq \frac{\sigma_N^2}{(2^{i-1} \Delta)^2} \implies i \geq \frac{1}{2} \log_2 \left( \frac{\sigma_N^2}{\Delta^2 P_e} \right) + 1$
Decoding and Correlation Tracking

Standard LMS

\[ u(n-1) \xrightarrow{\text{Adaptive Filter}} \hat{u}(n) \xrightarrow{\sum} u(n) \]

Decoding of compressed readings and correlation tracking

\[ x(n-1) \xrightarrow{\text{Adaptive Filter}} y(n) \xrightarrow{\text{DISCUS Decoder}} x(n) \]

\[ B(n) \]

\[ i(n+1) \]

\[ B(n) = \text{decoded readings of all other sensors} \]

\[ c(n) = \text{coset index of } x(n), \text{ sent by encoder of } x \]

\[ i(n+1) = \text{number of bits to use in encoding } x \text{ at time } n+1, \text{ fed back to encoder of } x \]
Adaptation Algorithm

- $U(n) = M \times 1$ input at time $n$
- $y(n) = W(n)' \ast U(n)$
- Use DISCUS decoding to find $x(n)$
- $e(n) = x(n) - y(n)$
- $W(n+1) = W(n) + \mu e(n) u(n)$
Experimental Setup

- Collected data from PicoRadio test-bed nodes
- 5 light, 5 temperature, 5 humidity sensors
- Data was collected and used for testing real-time algorithms
Simulations (correlation tracking)

- **Avg. Temp Savings = 66.6%**
- **Avg. Humidity Savings = 44.9%**
- **Avg. Light Savings = 11.7%**
Compressing encrypted content without the cryptographic key
Secure multimedia for home networks

- Uncompressed encrypted video (HDCP protocol)
  - Can increase wireless range with lower data rate
  - But how to compress encrypted video without access to cryptographic key?
Application: Compressing Encrypted Data

Traditional/Best Practice:

Message Source \((X)\) → Compression → Encryption → Public Channel → Decryption → Secure Channel → Decompression → Reconstructed Source

- Eavesdropper
- Key \((K)\)

Novel Structure:

Message Source \((X)\) → Encryption → Public Channel → Joint Decompression and Decryption → Reconstructed Source

- Cipher Text \((Y)\)
- Compression → Secure Channel

- Eavesdropper
- Key \((K)\)

Johnson & Ramchandran (ICIP 2003), Johnson et. al (Trans. on SP, Oct. 2004)
Example

Original Image → Encrypted Image → Compressed Encrypted Image

Decoding compressed Image → Final Reconstructed Image

10,000 bits → 5,000 bits

Decoding Using DISCUS Framework (Key as Side Information)
Application: compressing encrypted data

Source Image → Encrypter → Encoder → Decoder → Decrypter → Decoded Image

10,000 bits

How to compress to 5,000 bits?

Key Insight!

Source Image

Encrypted Image

Decoded Image

Joint Decoder/Decrypter

Syndrome

Reconstructed Source

Key

Key
Illustration: coding in action

- Bits of Source Y
- Bits of Source K
- Y,K correlation
- LDPC code applied to Y
- Compressed bits of Y
Overview

1. \( Y = X + K \) where \( X \) is indep. of \( K \)
2. Slepian-Wolf theorem: can send \( X \) at rate \( H(Y|K) = H(X) \)
3. Security is not compromised!

Framework: Encryption

Encryption:

- **Stream cipher**

  \[ y_i = x_i \oplus k_i \]

- **Graphical model**
  captures exact encryption relationship
Source Models

- **IID Model**

- **1-D Markov Model**

- **2-D Markov Model**
Encrypted image compression results

- 100 x 100 pixel image (10,000 bits)
- No compression possible with IID model

1-D Markov Source Model

Source Image  Encrypted Image  Compressed Bits  Decoded Image

2-D Markov Source Model

Rate = 0.77

Rate = 0.43
Key problems

- Data \( X \sim p(X) \)

- When source statistics \( p(X) \) are unknown
  - How to learn how much to compress?
  - How fast can limits be learned?

- When source statistics \( p(X) \) are known
  - How to develop practical compression codes?
  - How well can they perform?
“Blind” compression protocol

- For development: $X$ is IID, $X \sim Bernoulli(Q)$
- Blocks indexed by $i$
- Encoder uses source estimate $\hat{Q}_i, \hat{Q}_0 = 0.5$
- Compressible: $R = H(X) = H(Q)$
Sample run

**ENCODER**

<table>
<thead>
<tr>
<th>BITS</th>
<th>RATE</th>
<th>FEEDBACK</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>ACK, $\hat{Q}_1$</td>
</tr>
<tr>
<td>$R_2$</td>
<td></td>
<td>ACK, $\hat{Q}_2$</td>
</tr>
<tr>
<td>$R_3$</td>
<td></td>
<td>NAK</td>
</tr>
<tr>
<td>$R_4$</td>
<td>1</td>
<td>ACK, $\hat{Q}_3$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ACK, $\hat{Q}_4$</td>
</tr>
</tbody>
</table>

**DECODER**

- $\hat{Q}_1$
- $\hat{Q}_2$
- NAK
- $\hat{Q}_3$
- $\hat{Q}_4$
Rate selection

- Use $\epsilon$ redundancy to account for error in $Q$ estimate
Choose $\varepsilon_i$ parameter to minimize redundancy.
- Large $\rightarrow$ Low $\Pr(error)$, High $R_i$
- Small $\rightarrow$ High $\Pr(error)$, Low $R_i$

Define $c_i$ as the transmission rate of block $i$.

Minimize expected total rate: $E[c_i] = \Pr[e_i] \ln(2) + E[R_i]$

Intuition — $\varepsilon_i$ must go down $\sim 1/\sqrt{l}$

- $\hat{Q}_i = \sum_{j=1}^{n_i} x_j / n_i$
- $E[\hat{Q}_i] = Q, \quad \sigma_{\hat{Q}_i} = \sqrt{Q(1-Q)} / \sqrt{n_i}$
Analysis assumptions and bound

- Decoding errors are detected
- Decoding errors occur when empirical entropy exceed code rate (perfect codes)

\[ H(x_{n_{i-1}+1}^{n_i}) > l_i R_i \]

Resulting Bound Form: \[ E[c_i] \leq H(Q) + C_1 \varepsilon_i + C_2 \exp\{-C_3 l_i \varepsilon_i^2\} \]
- Linear term → Make \( \varepsilon \) small
- Exponential term → \( \varepsilon \) decays slower than \( 1/\sqrt{l_i} \)
Results – Memoryless Source

- $H(Q_1 = .027) = 0.17912$
- Uncompressed transmission (top line)
- With knowledge of source statistic $Q$ (bottom line)
Temporal Prediction Scheme

Decoded Frame n-3 ➔ Frame Predictor ➔ Predicted Frame n-1 ➔ Predictor Quality Measure
Decoded Frame n-2 ➔ Frame Predictor ➔ Predicted Frame n ➔ Decoder
Decoded Frame n-1 ➔ Actual Frame n

Predicted Frame n ➔ Key ➔ Encrypter ➔ Encrypted Frame n

Compressed Bits ➔ Encoder ➔ Decoded Frame n-3, Decoded Frame n-2, Decoded Frame n-1, Actual Frame n
Compression of encrypted video

- Video offers both temporal and spatial prediction
- Decoder has access to unencrypted prior frames

<table>
<thead>
<tr>
<th></th>
<th>Blind approach (encoder has no access to key)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foreman</td>
<td>Saves 33.00%</td>
</tr>
<tr>
<td>Garden</td>
<td>Saves 17.64%</td>
</tr>
<tr>
<td>Football</td>
<td>Saves 7.17%</td>
</tr>
</tbody>
</table>
Distributed Video Compression
Active research area

- Puri and KR: Allerton’02, ICASSP’03, ICIP’03.
- Aaron, Zhang and Girod: Asilomar’02
- Aaron, Rane, Zhang and Girod: DCC’03
- Aaron, Setton and Girod: ICIP’03
- Sehgal, Jagmohan and Ahuja: DCC’03, ICIP’03.
- Wang, Majumdar and KR: ACM MM’04
- Yaman and AlRegib: ICASSP’04
- Xu & Xiong: VCIP ’04
- Wang & Ortega: ICIP ’04

First side-information-coding based video coding idea was however in 1978!!
Application scope

- Motivation: Uncertainty in the side information
  - Low complexity encoding
  - Transmission packet drops
  - Multicast & scalable video coding
  - Flexible decoding

- Physically distributed sources
  - Multi-camera setups

- Other interesting applications?
Low complexity encoding

**Motivation**

- Low-Complexity Encoder
- Low-complexity encoding
  - No motion search
  - Various channel codes

**Trans-coding proxy**

- High-Complexity Decoder
- High-Complexity Encoder

**DSC Encoder**

- Current frame

**DSC Decoder**

- Current frame

**S-I Generator**

- Reference frame

- High-complexity
  - Block or frame-level
  - Interpolated or compensated motion

---

_Puri & Ramchandran, Allerton 2002_
_Aaron, Zhang & Girod, Asilomar 2002_
_Artigas, Ascenso, Dalai, Klomp, Kubasov & Ouaret, PCS 2007_
Transmission packet loss

- FEC solutions may be inadequate
- Can be made compatible with existing codec
- Corrupted current frame is S-I at DSC robustness decoder

A. Aaron, S. Rane, D. Rebollo-Monedero & B. Girod: DCC’03, ICIP’04, ICIP’05
A. Sehgal, A. Jagmohan & N. Ahuja: Trans’04
B. J. Wang, A. Majumdar, K. Ramchandran & H. Garudadri: PCS’04, ICASSP’05
Multicast & scalable video coding

- Multicast
  - Accommodate heterogeneous users
    - Different channel conditions
    - Different video qualities (spatial, temporal, PSNR)

Majumdar & Ramchandran, ICIP 2004
Tagliasacchi, Majumdar & Ramchandran, PCS 2004
Sehgal, Jagmohan & Ahuja, PCS 2004
Wang, Cheung & Ortega, EURASIP 2006
Xu & Xiong, Trans. Image Processing 2006
Flexible decoding

- \{Y_1, Y_2, ..., Y_N\} could be
  - Neighboring frames in time
    \rightarrow forward/backward playback without buffering
  - Neighboring frames in space
    \rightarrow random access to frame in multi-view setup
  - ...

Draper & Martinian, ISIT 2007
Multi-camera setups

- Dense placement of low-end video sensors
- Sophisticated back-end processing
  - 3-D view reconstruction
  - Object tracking
  - Super-resolution
- Multi-view coding and transmission
Other applications?

- Rate-efficient camera calibration
  - Visual correspondence determination

Tosic & Frossard, EUSIPCO 2007
Yeo, Ahammad & Ramchandran, VCIP 2008
PRISM: DSC based video compression

- Motivation:
  - Low encoding complexity
  - Robustness under low latency
A closer look at temporal aspects of video

- Motion is a “local” phenomenon
- Block-motion estimation is key to success
DFD Statistics: mixture process
DFD Statistics: mixture process

Prediction Information

Independent

DFD “innovations”

$\mathbb{Z}_2$

Current frame
DFD Statistics: mixture process

Prediction Information

Independent DFD “innovations”

\[ Z_i \]

Current frame
Frame-level treatment of DFD ignores block-level statistical variations.

Suggests block-level study of side-information coding problem

Challenge: How to approach MCPC-like performance with/without doing motion search at the encoder?
MCPC: a closer look

Motion search region for block $X$.

M candidate motion vectors.

Previous decoded frame.

Independent DFD “innovations” $Z$.

Current frame.

Motion-compensated prediction $Y_T$.

Prediction error (DFD) $Z$.
Motion-free encoding?
Motion-free encoding?

- The encoder does not have or cannot use $Y_1, \ldots, Y_M$
- The decoder does not know $T$
**Motion-free encoding?**

- The encoder does not have or cannot use $Y_1, ..., Y_M$.
- The decoder does not know $T$.
- The encoder may work at rate: $R(D) + (1/n) \log M$ bits per pixel.
- How to decode and what is the performance?
Let’s cheat!

- Let’s cheat and let the decoder have the motion vector $T \to \text{“classical” Wyner-Ziv problem}$
- The encoder works at same rate as predictive coder
What if there is no genie?

- Can decoding work without a genie?
  - Yes
- Can we match the performance of predictive coding?
  - Yes (when DFD statistics are Gaussian)

---

Ishwar, Prabhakaran, and KR, ICIP '03.
Source Encoding with side-information under Ambiguous State Of Nature (SEASON)

- The encoder does not have $Y_1, \ldots, Y_M$
- Neither the decoder nor the encoder knows $T$
- The MSE is still the same as for the MCPC codec

**Theorem:** SEASON codecs have the same rate-distortion performance as the MCPC codec for Gaussian DFD, i.e.

$$R(D) = \frac{1}{2} \log (\sigma_Z^2/D)$$
Source Encoding with side-information under Ambiguous State Of Nature (SEASON)

- **Theorem**: SEASON codecs have the same rate-distortion performance as the genie-aided codecs. This performance is (in general)

\[
R(D) = \min_{U} I(U;X) - I(U;Y_T)
\]

subj. \(X \leftrightarrow Y_T \leftrightarrow U\) is Markov, and \(f(\cdot, \cdot)\) such that \(E[(X-f(Y_T, U))^2] = D\)
Practical implementation

- Low-complexity motion-free encoder
- Can be realized through decoder motion search
  - Need mechanism to detect decoding failure
  - In theory: joint typicality (statistical consistency)
  - In practice: Use CRC
Noisy channel: drift analysis

**MCPC:** Channel errors lead to prediction mismatch and drift.

**PRISM:** Drift stopped if syndrome code is “strong enough”

- All that matters:
  - Targeted syndrome code noise \( \geq \)
  - Video innovation + Effect of Channel + Quantization Noise
Results

- Qualcomm’s channel simulator for CDMA 2000 1X wireless networks

- Stefan (SIF, 2.2 Mbps, 5% error)
Challenges: correlation estimation

Recall

\[ X \xrightarrow{\text{Encoder}} X' \xrightarrow{\text{Decoder}} \hat{X} \]

- **PRISM:**
  - Superior performance over lossy channels.
  - But compression efficiency inferior to predictive codecs.

- **Challenge:** correlation estimation, i.e. finding \( H(X|Y) = H(N) \)

- \( N = \text{Video innovation} + \text{Effect of channel} + \text{Quantization noise} \)
  - Hard to model without motion search
  - Without accurate estimate of the total noise statistics, need to over-design \( \rightarrow \) compression inefficiency.

- What if complexity were less of a constraint and we allow motion search at the encoder?
What if complexity were less of a constraint?

- Allow motion search at encoder → can model video innovation

- Distributed Video Coding can approach the performance of predictive coders when it estimates the correlation structure accurately

- How to enhance robustness by considering effect of channel?
Modeling effect of channel at enc.: finding $H(X | Y')$

- Efficient strategy to exploit natural diversity in video data

  - Encoder has access to both $Y$ and $Z$

  - Fact: there is natural **diversity** in video data
    - An *intact* second best predictor ($P_2$) is typically a better predictor than a *corrupted* best predictor ($P_1$)
    - Can be viewed as motion search with two candidates.

- The decoder **knows** to use the better of $P_1$ or $P_2$ as SI.
- We have control over uncertainty set at decoder

---

J. Wang, V. Prabhakaran & KR: ICIP’06
Finding $H(X|Y')$

- If we have some knowledge about the channel:
  
  $$Y' = \begin{cases} 
  Y & \text{if } Y \text{ is intact} \\
  Z & \text{if } Y \text{ is corrupted}
  \end{cases} \quad \text{with probability } (1-p)$$

- We obtain
  
  $$H(X|Y', \text{decoder state}) = (1-p)H(X|Y) + pH(X|Z)$$
Another way to think about it

- \[ H(X|Y', \text{decoder state}) = (1-p)*H(X|Y) + p*H(X|Z) \]

\[ = p*[H(X|Z) - H(X|Y)] + H(X|Y) \]

- Effect of channel
- Video innovation
Yet another way to think about it

- \( H(X|Y', \text{decoder state}) = (1-p)H(X|Y) + pH(X|Z) \)

Can be achieved by applying channel code to sub-bin indices

Additional syndrome (sub-bin index) for drift correction

Bare minimum syndrome (bin index) needed when channel is clean
Robustness result

Setup:

- Channel:
  - Simulated Gilbert-Elliot channel with $p_g = 0.03$ and $p_b = 0.3$
Robustness result

Setup:

- **Channel:**
  - Simulated CDMA 2000 1x channel

Stefan (SIF) sequence
1 GOP = 20 frames
1 mbps baseline, 1.3 mbps total (15 fps)
7.1% average packet drop rate

Football (SIF) sequence
1 GOP = 20 frames
900 kbps baseline, 1.12 mbps total (15 fps)
7.4% average packet drop rate
Videos

- **Garden**
  
  352x240, 1.4 mbps, 15 fps, gop size 15, 4% error  
  (Gilbert Elliot channel with 3% error rate in good state and 30% in bad state)

- **Football**
  
  352x240, 1.12 mbps, 15 fps, gop 15, simulated CDMA channel with 5% error
DSC for multi-camera video transmission
Distributed multi-view coding

- Encoder 1
- Encoder 2
- Encoder 3
- Channel
- Joint Decoder

- Video encoders operate independently
- Feedback possibly present

- Video decoder operates jointly
Active area of research

- **Distributed multi-view image compression**
  - Down-sample + Super-resolution [Wagner, Nowak & Baraniuk, ICIP 2003]
  - Geometry estimation + rendering [Zhu, Aaron & Girod, SSP 2003]
  - Direct coding of scene structure [Gehrig & Dragotti, ICIP 2005] [Tosic & Frossard, ICIP 2007]
  - Unsupervised learning of geometry [Varodayan, Lin, Mavlankan, Flierl & Girod, PCS 2007]
  - ...

- **Distributed multi-view video compression**
  - Geometric constraints on motion vectors in multiple views [Song, Bursalioglu, Roy-Chowdhury & Tuncel, ICASSP 2006] [Yang, Štankovic, Zhao & Xiong, ICIP 2007]
  - Fusion of temporal and inter-view side-information [Ouaret, Dufaux & Ebrahimi, VSSN 2006] [Guo, Lu, Wu, Gao & Li, VCIP 2006]
  - MCTF followed by disparity compensation [Flierl & Girod, ICIP 2006]
  - ...

- **Robust** distributed multi-view video compression
  - Disparity search / View synthesis search [Yeo, Wang & Ramchandran, ICIP 2007]
Robust distributed multi-view video transmission

Video encoders operate independently and under complexity and latency constraint.

Video decoder operates jointly to recover video streams
Side information from other camera views

\[ X = \text{Frame } t \]

\[ \hat{X} = \text{reconstructed Frame } t \]

- **How should we look in other camera views?**
  - Naïve approach of looking everywhere can be extremely rate-inefficient

- **Possible approaches**
  - View synthesis search
  - Disparity search

\[ Y' = \text{corrupted Frame } t-1 \]

\[ Y'' = \text{neighboring Frame } t \]
Epipolar geometry

- Given an image point in one view, corresponding point in the second view is on the epipolar line.

- Upshot: **Disparity search is reduced to a 1-D search**.
Decoder disparity search

Camera 1
Temporal – Poor reference

Camera 2
Spatial – Good reference

Frame \( t-1 \)  Frame \( t \)

- Extension of decoder motion search using epipolar geometry

\[
X = Y_{DS} + N_{DS}
\]

[Yeo & Ramchandran, VCIP 2007]
PRISM-DS vs MPEG with FEC

- “Ballroom” sequence (from MERL)
  - 320x240, 960 Kbps, 30fps, GOP size 25, 8% average packet loss

- Drift is reduced in PRISM-DS

[Yeo & Ramchandran, VCIP 2007]