From Conventional to Distributed Video Coding

Lina  Karam
Department of Electrical Engineering
Arizona State University
Tempe, AZ 85287
karam@asu.edu
Outline

Why Image and Video Compression?

Basics of Image and Video Signals

• Basic imaging system
• Digitization: sampling and quantization
• Redundancy and correlation
Outline

Conventional Video Compression

• Basic system
• Discrete-Cosine Transform (DCT)
• Predictive Coding
• Motion estimation and compensation
• Entropy Coding
• H.264/AVC (MPEG 4 Part 10)
Outline

Distributed/Wyner-Ziv Video Coding
  • Motivation
  • Slepian-Wolf and Wyner-Ziv Coding
  • Basic Distributed Video Coding Framework
  • Existing Approaches
  • Proposed BLAST-DVC
  • Results

Conclusion
Outline

Why Image and Video Compression?

Basics of Image and Video Signals

• Basic imaging system
• Digitization: sampling and quantization
• Redundancy and correlation
Need for Compression

Why Image and Video Compression?

- Real-time video runs at 30 frames per second – need to transfer each frame in 30 msec.
  - Wireless (384Kbps): 6.4 second for a single frame
  - Cable Regular
    Upload (256 Kbps): 9.5 s/frame
    Download (1.5Mbps): 1.6 s/frame
  - Cable Premium
    Upload (1 Mbps): 2.4 s/frame
    Download (12 Mbps): 200 msec/frame
  - Low Power, Harsh Environments: 122 secs/frame
Outline

Why Image and Video Compression?

Basics of Image and Video Signals

• Basic imaging system
• Digitization: sampling and quantization
• Redundancy and correlation
Colored lights from scene are captured into red, green, and blue pixels (picture elements)

Scene viewed through “color” filters that separate the image into 3 color components

Digital camera systems contain optics that image light onto sensors typically a CCD array with filters
Outline

Why Image and Video Compression?
• Sample image-and video-based applications
• Need for compression

Basics of Image and Video Signals
• Basic imaging system
• Digitization: sampling and quantization
• Redundancy and correlation
Digitization: Sampling and Quantization

- Original Imaged Scene: analog (continuous in space and time) $I(x,y;t)$ for video and $I(x,y)$ for still image
  
  $I$: image intensity and color at position $(x,y)$ and at time $t$

- Digitized sensed image/video: digital (sampled in space and time, plus discrete amplitudes) $I(n_1,n_2,n_3)$ for video and $I(n_1,n_2)$ for still image

  $I$: image intensity and color at integer sample position $(n_1,n_2)$ and integer time index $n_3$
Digitization: Sampling and Quantization

Video Sampling

• Temporal sampling affects frame (image) rate and perceived motion quality.
  – 50 to 60 frames per second produce smooth apparent motion
  – 25 (PAL) or 30 (NTSC) frames per second is standard for television pictures; interlacing can be used to improve the appearance of motion

• Frame rate can be referred to as temporal resolution.
Quantization

Some sort of quantization is necessary to represent continuous signals in digital form

Quantization is also used for data reduction in virtually all lossy coding schemes
Quantization – Basic Concepts

Two main types of quantizers:

• **Scalar Quantizer (SQ)**
  - Operates on each data value individually
  - Data range is divided into \( M \) non-overlapping bins
  - Each input data value is assigned to one of the bins
  - Output of quantizer is index of assigned bin

• **Vector Quantizer (VQ)**
  - Operates on blocks of values
  - Data space is divided into bins
  - Each block of values is assigned to a bin
  - Output of VQ is index of the assigned bin

Uniform Quantizer \( \Rightarrow \) equal-width bins

Step-Size \( \Delta = \text{bin width} \)
\[
\Delta = \frac{(I_{\text{max}} - I_{\text{min}})}{M}
\]

Very large computational complexity and memory requirements

Copyright 2007 by Lina J. Karam
Outline

Why Image and Video Compression?
• Sample image-and video-based applications
• Need for compression

Basics of Image and Video Signals
• Basic imaging system
• Digitization: sampling and quantization
• Redundancy and correlation
Redundancy and Correlation

Significant redundancy present in natural image/video signals.

Redundancy is proportional to the amount of correlation present in the image data samples.

Types of redundancy:

• Spatial: interpixel redundancy
• Temporal: interframe redundancy
• Statistical: data distribution, frequency of occurrence
Redundancy and Correlation: Spatial
Redundancy and Correlation: Temporal
Redundancy and Correlation: Temporal

Mother and Daughter
CIF – 352 x 288

Frame 60

Frame 61

Time
Redundancy and Correlation: Statistical

Statistical: data distribution, frequency of occurrence

\[ P(I) \]

\[ P(I) \]

Copyright 2008 by Lina J. Karam

From Conventional to Distributed Video Coding
Outline

Conventional Video Compression

• Basic lossy compression system
• Basic video compression system
• Discrete-Cosine Transform (DCT)
• Predictive Coding (DPCM)
• Motion estimation and compensation
• Entropy Coding
• H.264/AVC (MPEG 4 Part 10)
Basic Lossy Compression System

A form of data compression; usually lossless, but can be lossy

Lossless compression:
• Transform Coding
• Predictive Coding
• Motion Prediction
• Sub-band Coding, Wavelets
• Run-Length Coding

Lossy compression; typically removes less significant or irrelevant information

Lossless coding:
• Fixed-Length Coding
• Variable-Length Coding, Entropy Coding

Perform inverse operations at the receiver / decoder

Image or Video

01001100...
Bit Stream

Transmission or Storage
Outline

Conventional Video Compression

• Basic lossy compression system
• Basic video compression system
• Discrete-Cosine Transform (DCT)
• Predictive Coding (DPCM)
• Motion estimation and compensation
• Entropy Coding
• H.264/AVC (MPEG 4 Part 10)
Basic Lossy Compression System

From Conventional to Distributed Video Coding
Outline

Conventional Video Compression

- Basic lossy compression system
- Basic video compression system
- Discrete-Cosine Transform (DCT)
- Predictive Coding (DPCM)
- Motion estimation and compensation
- Entropy Coding
- H.264/AVC (MPEG 4 Part 10)
Redundancy Reduction – Transform Coding

$N_1 \times N_2$ Discrete Cosine Transform – DCT (Type IV, Unitary)

$$W_{DCT}(K_1, K_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} I(n_1, n_2) C_1(K_1) C_2(K_2) \cos \left( \frac{\pi (2n_1 + 1)}{2N_1} K_1 \right) \cos \left( \frac{\pi (2n_2 + 1)}{2N_2} K_2 \right)$$

DCT Basis images

where

$$C_i(K_i) = \begin{cases} \frac{1}{\sqrt{N_i}}; & K_i = 0 \\ \sqrt{\frac{2}{N_i}}; & K_i \neq 0 \end{cases} \quad ; \quad 0 \leq K_i \leq N_i - 1; \quad i = 1, 2$$
Redundancy Reduction – Transform Coding

Why use DCT?

Energy Conservation: $150^2 = 4 (75)^2 = 22,500$

Energy Compaction

Unwrap

Unwrap

Copyright 2008 by Lina J. Karam

From Conventional to Distributed Video Coding
Redundancy Reduction – Transform Coding

Why use DCT?

• Excellent energy compaction achieved for natural images
• Values in image transformed into few non-zero weights
• The non-zero weights are clustered at and near “low frequencies”

![Image of a woman with a hat]

2D DCT magnitude

\[ W(K_1, K_2) \]

\[ N_1 \times N_2 \]

\[ K_2 \]

\[ N_2 \]

\[ K_1 \]

\[ N_1 \]
Redundancy Reduction – Transform Coding

DCT applied to image blocks typically of size 8x8 or size 4x4

- Can exploit better local image characteristics
- Can exploit local correlations between neighboring image samples

Image block $I(n_1, n_2)$
Outline

Conventional Video Compression

• Basic lossy compression system
• Basic video compression system
• Discrete-Cosine Transform (DCT)
• Predictive Coding (DPCM)
• Motion estimation and compensation
• Entropy Coding
• H.264/AVC (MPEG 4 Part 10)
Redundancy Reduction – Predictive Coding

Predictive Coding

• Redundancy related to factors such as predictability, randomness, smoothness in the data
• Examples of predictive coding methods:
  – Linear prediction (DPCM)
  – Motion estimation and compensation
Redundancy Reduction – Linear prediction

Predictive coding can be applied to either the spatial or transform domain.

Basic concept:

• Remove mutual information between successive or adjacent data (pixels, blocks, frames) using prediction

• Given past samples (input or decoded), predict the subsequent data value using a prediction rule

\[ I_p(n) = P[I(n-1), I(n-2), \ldots, I(n-M)] \]

• Encode only the new information: code difference between actual and predicted

\[ e_p(n) = I(n) - I_p(n) \]
Redundancy Reduction – Linear prediction

Linear prediction => Express predicted value as a weighted sum of past values
\[ I_p(n) = w_1 I(n-1) + w_2 I(n-2) + ... + w_M I(n-M) \]
Weights \( w_1, \ldots, w_M \) are found to minimize prediction error norm based on correlation in the image data

Order \( M \) of predictor chosen to be relatively small
  - Computations increase as \( M \) increases
  - Storage increases as \( M \) increases
  - Prediction does not improve much with large \( M \)

Coding process done recursively
Actual value can be losslessly recovered: \( I(n) = I_p(n) + e_p(n) \)
Redundancy Reduction – Linear prediction

Consider an image row

\[ I_p(n) = I(n-1) \]

Prediction error is:

\[ e_p(n) = I(n) - I(n-1) \]

Divide into \( L \) levels

\( B \) bits – \( L = 2^B \) levels

\( \Rightarrow \) Higher accuracy and less distortion if we are limited to \( B \) bits
Redundancy Reduction – Linear prediction

Issue when prediction error $e_p(n)$ has to be limited in accuracy (loss due to quantization)

- Error $e_p(n)$ is not coded losslessly but approximated with $\hat{e}_p(n)$
- At receiver, $\hat{I}(n)$ is obtained instead of actual value $I(n)$

$$\hat{I}(n) = \hat{I}_p(n) + \hat{e}_p(n)$$

Predictions at receiver computed from distorted values $\hat{I}(n-1), \hat{I}(n-2), ..., \hat{I}(n-M)$ ⇒ receiver/decoder not synchronized with encoder ⇒ this leads to unbounded error accumulation (unstable system) in reconstructed values

Need to synchronize the encoder and decoder by predicting from past decoded values and not actual input samples
Redundancy Reduction – Linear prediction

Feedforward Prediction

• Coder:

\[ \hat{e}_p(n) = e_p(n) - I_p(n) \]

I(n) is used to predict

• Decoder

\[ \hat{I}(n) = \hat{e}_p(n) - I(n) \]

⇒ Unstable system due to error accumulation since reconstruction error \( \hat{I}(n) - I(n) \) would depend on all past and present quantization errors
Redundancy Reduction – Linear prediction

Feedback Prediction: also referred to as Differential Pulse Coded Modulation (DPCM)

- Encoder:

- Decoder

⇒ Stable system: reconstruction error depends only on present quantization error \( \delta \) ⇒ \( \hat{I}(n) - I(n) = \delta \)
Redundancy Reduction – Linear prediction

For the two-dimensional (2D) case $I(n_1,n_2)$

- Two-dimensional DPCM
- Previous 1-D discussion extends to 2-D case
- For common (natural) images, it was found that restricting the prediction to the four nearest (causal) neighbors is sufficient $\Rightarrow$ four non-zero prediction weights

- Example: row-by-row scanned image

\[
A = w_1 B + w_2 C + w_3 D + w_4 E
\]
Outline

Conventional Video Compression

• Basic lossy compression system
• Basic video compression system
• Discrete-Cosine Transform (DCT)
• Predictive Coding (DPCM)
• Motion estimation and compensation
• Entropy Coding
• H.264/AVC (MPEG 4 Part 10)
Motion Estimation and Compensation

Video is a sequence of frames (images)

A typical scene does not change rapidly from one frame to the next.

Huge amount of data that need to be transmitted and played-back in real time or fast enough.
Motion Estimation and Compensation

Exploit intraframe (within one image) redundancy and interframe (between frames) redundancy (motion estimation and compensation).

Most common method to remove interframe redundancy is motion compensation (MC).

- Illustration with linear motion, black ball, white background.

\[
\begin{align*}
\vec{d} & \quad d_y \\
\vec{d} & \quad d_x \\
time & \\
\end{align*}
\]

- If one can extract the ball, just transmit how far it moved each frame (send a vector describing the displacement of the object, motion compensation vector).
Motion Estimation and Compensation

Applications

• Compression
• Restoration
• Interpolation and increase in size of image or in frame rate

Motion estimation

• Motion estimation is the process of determining the movement of objects within a sequence of image frames.

Motion compensation

• Motion compensation refers to the generation of the predicted frame using the estimated motion.
Motion Estimation

Motion estimation techniques can be categorized into three major groups:

- Pixel-based techniques
  - Pel-recursive algorithms
  - Gradient-descent algorithms
- Block-based (Block Matching) techniques
  - Exhaustive full/fractional pel search
  - Fast search algorithms
  - Hierarchical block matching algorithms
- Object-based techniques
  - mesh-based techniques
  - segmentation-based techniques
Motion Estimation

Motion estimation based on Block Matching is commonly used and is incorporated within the video coding standards.

Basic procedure for block matching

1. Divide adjacent and current frames into small blocks.
2. For a given block in current frame, search for the displacement (translational motion) which produces the “best match” among possible blocks in an neighboring/adjacent frame.
Motion Estimation

Block matching-based (Block-based) motion compensation

1. Divide current and previous frame into non-overlapping blocks with size $N_1 \times N_2$.
2. For each block in current frame:
   1. Search in reference frame (previously encoded frame) for a block that closely matches the current block: search region in reference frame is centered at current block position and is of size $(2p+N_1) \times (2p+N_2)$.
   2. Find closest matching block with respect to a given error metric.
   3. Send motion vector $(m_x, m_y)$. 
Motion Estimation

- To perform search (Step 2.1), we search in reference frame for matching block by moving window (of same size as block) in pixels at a time (not in blocks at a time).
- Large blocks result in fewer blocks \(\Rightarrow\) fewer motion vectors \(\Rightarrow\) poor accuracy but low bit-rate
- Small blocks result in more blocks \(\Rightarrow\) good accuracy but high bit-rate
- Typically, for motion estimation, blocks (macroblocks) for MPEG standards are chosen to be \(16 \times 16\), but also can be with sizes smaller than that (8x8, 4x4, variable).
Motion Estimation

How “best match” block is characterized?

• \( B_t(n_1, n_2) : N_1 \times N_2 \) image block in current frame
• \( B_{t-l}(n_1, n_2) : N_1 \times N_2 \) image block in reference frame
• Compute prediction (match) error:
  \[ e_p(n_1, n_2) = B_t(n_1, n_2) - B_{t-l}(n_1, n_2) \]
• Compute error norm \( \Rightarrow \) gives one score
  – Mean-Squared Error (MSE) – \( L_2 \) norm
    \[ \left\| e_p(n_1, n_2) \right\|^2 = \frac{1}{N_1 N_2} \sum_{n_1} \sum_{n_2} e^2_p(n_1, n_2) = \frac{1}{N_1 N_2} \sum_{n_1} \sum_{n_2} (B_t(n_1, n_2) - B_{t-l}(n_1, n_2))^2 \]
  – Sum of Absolute Difference (SAD) – \( L_1 \) norm
    \[ \left\| e_p(n_1, n_2) \right\|_1 = \sum_{n_1} \sum_{n_2} \left| e_p(n_1, n_2) \right| = \sum_{n_1} \sum_{n_2} \left| B_t(n_1, n_2) - B_{t-l}(n_1, n_2) \right| \]
Motion Estimation

Fractional Pixel (Sub-pixel) Motion Estimation

• Earlier block-matching scheme limited to integer (full-pixel) displacement
• Better accuracy achieved if fractional pixel displacement can be detected
• Half-pixel (half-pel) displacement accuracy can be achieved by interpolating (enlarging) the reference frames by two in each of the horizontal and vertical directions
• Quarter –pixel (quarter-pel) displacement accuracy can be achieved by interpolating the reference frames by four in each of the horizontal and vertical directions
Motion Estimation

Half-Pixel Block-based Motion Estimation

Interpolate the reference frame by two in width and height
Perform search on the reference frame
Better accuracy at the expense of higher computations (up to four times more computations than full-pixel accuracy)
Motion Estimation and Compensation

Reference Frame \textit{(Frame 13)}

Current Frame \textit{(Frame 14)}
[8x8] block motion vectors superimposed on Reference F[n-1]  (Frame 13)
Motion Compensated Reference $F[n-1]$  \hspace{1cm} (Frame 14)

$PSNR = 34\ dBs$, $MSE = 24.5$
Motion Compensated Residual Frame
Residual Frame No Motion Compensation
Motion vector map (16x16 blocks, Full pixel)
Motion vector map (4x4 blocks, Full pixel)
Residual Frame No Motion Compensation
Residual Error (16x16 blocks, Full pixel)

\[ \text{PSNR} = 31.6 \text{ dB}, \text{ MSE} = 45.1 \]
Residual Error (8x8 blocks, Full pixel)

\[ \text{PSNR} = 34.2 \text{ dB, MSE} = 24.5 \]
Residual Error (4x4 blocks, Full pixel)

PSNR = 37.3 dB, MSE = 12
Residual Error (4x4 blocks, Half pixel)

PSNR = 39.5 dB, MSE = 7.5
Residual Error (4x4 blocks, quarter pixel)

$PSNR = 40.1\, dB$, $MSE = 6.3$
Motion Estimation and Compensation

- I frame: intra frame (also called key frame) is intra coded
- P frame: forward predicted frame from one previously decoded frame
- B frame: bi-directional predicted frame from one or two previously coded frames
- Group of Pictures (GoP): I BB P BB P BB P BB ... P BB I
Outline

Conventional Video Compression

- Basic lossy compression system
- Basic video compression system
- Discrete-Cosine Transform (DCT)
- Predictive Coding (DPCM)
- Motion estimation and compensation
- Entropy Coding
- H.264/AVC (MPEG 4 Part 10)
Information Theory Basics

Consider an information source that can generate $L$ messages: $M_1, M_2, ..., M_L$

These messages do not typically occur with the same frequency.

Each message $M_k$ ($k$ takes values between 1 to $L$) is assigned a probability of occurrence $P_k$, which indicates how often a message can occur relative to other messages

- $P_k$ is expressed as a percentage and takes values between 0 and 1 (100/100)
Information Theory Basics

The information associated with a message $M_k$ is defined as

$$I_k = -\log_2(P_k) \text{ (bits)} ; \quad P_k = \text{probability of message } M_k$$

- **Note 1:**
  \[\log_2(P_k = 0) = -\infty \Rightarrow I_k = \infty\]
  \[\Rightarrow I_k \text{ is large when an unlikely message is generated}\]

- **Note 2:**
  \[\log_2(P_k = 1) = 0 \Rightarrow I_k = 0\]
  \[\Rightarrow \text{Certain message contains no information}\]

- **Note 3:**
  \[\sum_{k=1}^{L} P_k = 1 \Rightarrow 0 \leq P_k \leq 1 \Rightarrow I_k \geq 0\]
Information Theory Basics

Entropy

- Measure of the average amount of information content in the signal (non-context info content)
- The first-order entropy (also called “entropy”) of a discrete information source that can generate $L$ messages $M_k$ with probability $P_k$, is:

$$H = \sum_{k=1}^{L} P_k I_k = -\sum_{k=1}^{L} P_k \log_2(P_k) \text{ bits/message (bits per message)}$$

Average information content generated by source

- $H$ is the absolute minimum (lower bound on) average bitrate (bits per message) that can be used to encode the message without loss under assumption that messages are uncorrelated (or no info about inter-message dependency is available)
Information Theory Basics

Entropy is maximum for uniform (flat) distributions (equally-likely occurrences)

\[ P_k = \frac{1}{L}; \quad k = 1, \ldots, L; \quad L = \text{Total number of messages} \]

\[ H = -\sum_{k=1}^{L} \frac{1}{L} \log_2 \frac{1}{L} = \log_2 L = \max_{p_k} H \]
Entropy Coding

Objective:
Use $n_k = -\log_2(P_k) = I_k$ bits to code message $M_k$

Result:
Average bit rate

$$B = \sum_{k=1}^{L} P_k n_k$$

is equal to the entropy $H$ if $n_k = -\log_2(P_k)$ $\Rightarrow$ lower bound is achieved (assuming independent samples).

Issue: integer number of bits assigned to message $\Rightarrow$ up to 1 bit above entropy.
Entropy Coding

Two widely-used techniques for entropy coding are:

– Huffman coding
– Arithmetic coding
Outline

Conventional Video Compression

• Basic lossy compression system
• Basic video compression system
• Discrete-Cosine Transform (DCT)
• Predictive Coding (DPCM)
• Motion estimation and compensation
• Entropy Coding
• H.264/AVC (MPEG 4 Part 10)
Outline

Distributed/Wyner-Ziv Video Coding

• Motivation and Objectives
• Slepian-Wolf and Wyner-Ziv Coding
• Basic Distributed Video Coding Framework
• Existing Approaches
• Proposed BLAST-DVC
• Results
Distributed Video Coding: Motivation

Conventional video coding
- MPEGx or H.26x
- High complexity video encoder due to motion estimation.

Emerging applications
- Video compression with mobile devices
  - Low complexity video encoder is preferred to reduce the hardware cost and to extend battery life.
- Video compression for sensor networks
  - Low complexity video encoder is also preferred to reduce the hardware cost and to extend battery life.
  - Inter-sensor communication may not be allowed or needs to be minimized.

Two main frameworks
- Multi-View/Multi-Cameras
- Single-View/Single Camera (Wyner-Ziv Video Coding)
Distributed Video Coding: Motivation
Distributed Video Coding: Motivation
Distributed Video Coding: Objectives

Intraframe encoding and interframe decoding

- Remove complexity (motion estimation) from encoder to decoder
- Achieve interframe compression rate-distortion performance

Distributed source coding

- Compress two consecutive frames separately
- Decode the frames jointly at the decoder.
Outline

Distributed/Wyner-Ziv Video Coding

• Motivation

• Slepian-Wolf and Wyner-Ziv Coding

• Basic Distributed Video Coding System

• Existing Approaches

• Proposed BLAST-DVC

• Results

Conclusion
Slepian-Wolf Theorem

$X$ and $Y$ are correlated two i.d sources. $X$ and $Y$ dependency is described by $P(x|y)$.

$X$ and $Y$ can be compressed independently and decompressed jointly with a bit rate that is the same as the one when the two sources are compressed and decompressed jointly.

$R_x + R_y \geq H(X,Y)$

$R_x \geq H(X \mid Y)$ and $R_y \geq H(Y \mid X)$

\[ RX + RY = H(X,Y) \]

Asymmetric case

Vanishing error probability for long sequences

No errors

[bits]

Source $X$ Encoder $X$

Source $Y$ Encoder $Y$

Joint Decoder

$X$

$Y$

$H(Y)$

$H(Y|X)$

$H(X|Y)$

$H(X)$

$H(X/Y)$

Copyright 2008 by Lina J. Karam

From Conventional to Distributed Video Coding
Wyner-Ziv Coding  

- $R_{X|Y}(D)$ denotes, for a distortion $D$, the rate that is required if the side information $Y$ is available at the encoder.

- $R_{WZ}(D)$ denotes, for a distortion $D$, the achievable lower bound for the bit rate if the side information $Y$ is not available at the encoder.

- Slepian-Wolf gives: $R_{WZ}(0) = R_{X|Y}(0) = H(X | Y)$

- In the case of Gaussian sources, Wyner-Ziv Theorem gives: $R_{WZ}(D) = R_{X|Y}(D)$
Outline

Distributed/Wyner-Ziv Video Coding

• Motivation
• Slepian-Wolf and Wyner-Ziv Coding
• Basic Distributed Video Coding Framework
• Existing Approaches
• Proposed BLAST-DVC
• Results

Conclusion
Basic Distributed Video Coding Framework

Video → Quantization → Slepian-Wolf Encoder → Slepian-Wolf Decoder → Reconstruction → Side Information → Decoded Video

Wyner-Ziv Encoder

Wyner-Ziv Decoder

Copyright 2008 by Lina J. Karam
Outline

Distributed/Wyner-Ziv Video Coding

• Motivation
• Slepian-Wolf and Wyner-Ziv Coding
• Basic Distributed Video Coding Framework
• Existing Approaches
• Proposed BLAST-DVC
• Results

Conclusion
Existing Approaches

• Syndrome
  - Linear block codes
  - LDPC codes
• Parity bits
  - Turbo codes
• Previous frames
  - Motion compensated extrapolation
  - Hash-code-assisted motion compensation
• Previous frames and future frames
  • Motion compensated interpolation
  • Side information refinement
• Pixel domain
• Transform domain

Copyright 2008 by Lina J. Karam
From Conventional to Distributed Video Coding
Existing Approaches

**PRISM (Puri et al., IEEE Trans. IP, Oct 2007)**

- After quantization, bitplane representation used
- Most significant bit can be inferred from side information $Y$ that is correlated with $X$
- Least significant bits cannot be inferred and need to be encoded and sent to decoder – these constitute the Wyner-Ziv encoding also called syndrome - each of these bits identifies a codeword partition or coset
- Number of least significant bits needed equal tree-depth for which the distance between successive codewords in the partition is greater than twice the correlation noise magnitude
Existing Approaches


![Diagram of PRISM](image)

Quantized version of $X_i$

$\delta = \text{quantization step-size}$

(Figure from *Puri et al.*, *IEEE TIP*, Oct 2007)
Existing Approaches

PRISM (Puri et al., IEEE Trans. IP, Oct 2007)

- Syndrome-based Wyner-Ziv Coding by dividing codeword space into cosets
- Entropy coding used for the low-significant bitplanes (syndromes)
- Since correlation not known in practice, motion estimation and compensation is used at decoder to find best side information block.
- Hash-code-assisted motion estimation used for error resilience when channel is noisy: cyclic redundancy check is used for hash.
- Issues:
  - Coding rate is fixed in advance and cannot be adapted
  - Coding can stop if CRC check fails
Existing Approaches


- Rate-Compatible Punctured Turbo Codes (RCPT)
  - Only subset of parity bits sent to decoder

- Hybrid FEC/ARQ–like scheme
  - Feedback channel is to acknowledge the decoding correctness and to request more parity bits if needed.

- Progressive bitplane decoding
  - Decoded bitplanes provide extra information for decoding.
Existing Approaches

Rate-Adaptive DVC with Feedback: Transform-domain

**Intraframe Encoder**

- Wyner-Ziv frames
- $s$
- DCT
- $2^{M_k}$ levels
- Quantizer
- Extract bitplanes
- Slepian-Wolf Encoder
- Buffer

**Interframe Decoder**

- Wyner-Ziv Encoder
- Side Information
- $q_k'$
- Slepian-Wolf Decoder
- Reconstruction
- $\hat{X}_k$
- DCT
- $X_k'$
- IDCT
- Decoded Wyner-Ziv frames
- $S'$

**Key frames**

- $K$
- Conventional Intraframe Encoder
- Conventional Intraframe Decoder
- Decoded Key frames

**Side Information**

- $\hat{S}$
- Side Information Generation

**Request bits**

- $K'$
Existing Approaches
Rate-Adaptive DVC with Feedback: Pixel-domain

Intraframe Encoder

Wyner-Ziv Frame Encoder

Extract bitplanes

Slepian-Wolf Encoder

Buffer

Request bits

Slepian-Wolf Decoder

Reconstruction

Side Information

Conventional Intraframe Encoder

Conventional Intraframe Decoder

Extra ct

bitplanes

Wyner-Ziv Frame Decoder

Decoded Wyner-Ziv frames

Decoded Key frames

Side Information Generation

Copyright 2008 by Lina J. Karam
Existing Approaches: Improvements on Rate-Adaptive DVC with Feedback

Side information generation
- Hash-code-assisted motion estimation by Aaron et al. 2004
- Smoothness filtering on motion vectors by Brites et al. 2005
- Fractional-pixel accuracy motion estimation by Klomp et al. 2006
- Three-Dimensional Recursive (3DRS) block matching, Chien et al., 2006
- Multi-reference frame motion estimation by Zhuo et al. 2007

Side information refinement
- Partial decoded frames with weighted forward motion estimation by Ascenso et al. 2005
- Partial decoded frames with bi-directional motion estimation by Adikari et al. 2006

Slepian-Wolf codec
- Low Density Parity Check Codes (LDPC), Schonberg et al., 2004
- LDPC with accumulates syndrome (LDPCA) codes, D. Varodayan et al., 2006
- Rateless LDPC codes, D. K. He et al., 2008.

Feedback channel removal
- Error probability of the original frame and the average of two neighboring Wyner-Ziv frames, Artigas et al., 2005
- Estimate error probability from the variance of the original frame and the average of two neighboring Key frames, Morbee et al., 2007
Outline

Distributed/Wyner-Ziv Video Coding

- Motivation
- Slepian-Wolf and Wyner-Ziv Coding
- Basic Distributed Video Coding Framework
- Existing Approaches
- Proposed BLAST-DVC
- Results

Conclusion
Proposed BLAST-DVC

• Existing DVC systems decode every bitplane of each pixel or DCT coefficient.
• Could we just decode some bitplanes or part of bitplanes that are more important than others?
• How do we measure the importance of a bitplane?
  ⇒ Strategy: estimate distortion-rate ratios $\Delta D/\Delta R$ at decoder
  ⇒ LDPCA decoding only applied to bitplanes with high distortion-rate ratios
• Distortion-Rate Ratio
  $\Delta D$: the reduction in distortion reduction if a bitplane is LDPCA decoded
  $R$: bits needed to decode a bitplane = CRC bits for error detection + parity bits for error correction (based on error probability)
Proposed BLAST-DVC

- Rate-distortion based selective bitplane decoding for pixel-domain DVC
- Parity bits placed locally and in an adaptive manner where they are needed most
- Any competitive side information generation scheme can be used with proposed scheme.
Proposed BLAST-DVC

Wyner-Ziv Frame Encoder

\[ X_i \rightarrow \text{Divide into Sub-images} \rightarrow X_{i,1}, X_{i,2}, \ldots, X_{i,M} \rightarrow \text{Extract bitplanes} \rightarrow X_{i,m,1}, X_{i,m,2}, \ldots, X_{i,m,k} \rightarrow \text{LDPCA Encoder} \rightarrow \text{Buffer} \rightarrow \text{Block Indices Decoding} \]

Wyner-Ziv Frame Decoder

\[ X_i' \rightarrow \text{Decoded Wyner-Ziv frames} \rightarrow \text{Merge Sub-images} \rightarrow X_{i,1}', X_{i,2}', \ldots, X_{i,M}' \rightarrow \text{Minimum-distortion Pixel Reconstruction} \rightarrow \text{Minimum Distance Symbol Reconstruction} \rightarrow \text{LDPCA Decoder} \rightarrow \text{Block Indices Encoding} \]

Copyright 2008 by Lina J. Karam

From Conventional to Distributed Video Coding
Proposed BLAST-DVC: Distortion-Rate Ratio Estimation

- The original source information is not available at the decoder, so the distortion cannot be exactly measured. The bitrate cannot be known without decoding.
- Distortion-Rate Ratio estimation performed at the decoder using the side information frames
  - The complexity of the encoder is not increased
  - More flexibility as the decoder can selectively decode the bitplanes based on a target distortion-rate ratios. The target rate-distortion ratio can be changed so that different R-D operating point can be achieved.
  - Error probability needs to be estimated at decoder
BLAST-DVC: Distortion-Rate Ratio Estimation

- Let $D$ be the difference of the source information $X$ and its side information $X_{side}$.

- $D$ can be modeled as a random variable with a Laplacian distribution.

  $$
P(D = d) = \begin{cases} 
  \int_{2545}^{\infty} 0.5 \cdot \alpha \cdot \exp(-\alpha|x|) \, dx, & \text{if } d=255 \\
  \int_{d-0.5}^{d+0.5} 0.5 \cdot \alpha \cdot \exp(-\alpha|x|) \, dx, & \text{if } -255<d<255, \\
  \int_{-\infty}^{-2545} 0.5 \cdot \alpha \cdot \exp(-\alpha|x|) \, dx, & \text{if } d=-255 
\end{cases}$$

- $\alpha$ can be estimated from the co-located blocks of two motion-compensated Key frames $\hat{X}_{i-1}$ and $\hat{X}_{i+1}$.

  $$\frac{2}{\alpha_m^2} = \hat{\sigma}_m^2 = \frac{1}{N} \sum_{n=1}^{N} \left( \hat{X}_{i+1,m,n} - \hat{X}_{i-1,m,n} \right)^2,$$

  where $m$ is the block index and $n$ is the pixel location in the block $m$. 

Copyright 2008 by Lina J. Karam

From Conventional to Distributed Video Coding
Side Information Generation

Motion Compensated Interpolation

- Forward Motion Compensation
- Backward Motion Compensation
- Average
- Key frame

- Side information

Copyright 2008 by Lina J. Karam
Gaussian low pass filter is applied to the decoded Key frames \(i-1\) and \(i+1\) to improve the motion estimation accuracy.

3-D-RS block matcher is used to estimate the motion vectors from frame \(i-1\) (reference) to frame \(i+1\) (anchor).

Motion refinement removes the rounding error of motion vectors introduced by the projection of the motion vector from \(i-1\) to \(i\).

Block erosion eliminates the discontinuity of two neighboring motion vectors.
BLAST-DVC: Error Probability Estimation

The probability of bit error can be expressed as:

\[
P_{n,k} = P(b_{n,k} = 1, b'_{n,k} = 0 \mid b'_{n,r} = b_{n,r}, r \in DBPs, 1 \leq r \leq k - 1) + P(b_{n,k} = 0, b'_{n,k} = 1 \mid b'_{n,r} = b_{n,r}, r \in DBPs, 1 \leq r \leq k - 1)
\]

where \( b_{n,k} \) and \( b'_{n,k} \) denote a bit in the \( k \)th bitplane corresponding to the \( n \)th pixels in the original subimage and in the side information (generated through motion compensated interpolation), respectively. DBP stands for Decoded Bit Planes.
BLAST-DVC: Error Probability Estimation

Assume first bit of pixel correctly decoded as 0 and consider second bit.

\[ P(X | X_{side}) \]

\[ P_e = P_{n,k} = P(b_{n,k} = 0 | X_{side}) = \frac{P_1}{P_1 + P_2}, \quad P(b_{n,k} = 1 | X_{side}) = \frac{P_2}{P_1 + P_2} \]
BLAST-DVC: Error Probability Estimation

Assume first bit of pixel not decoded (could be 0 or 1) and consider second bit.

\[ P(X \mid X_{side}) \]

\[ P_e = P_{n,k} = P(b_{n,k} = 0 \mid X_{side}) = P_1 + P_3, \quad P(b_{n,k} = 1 \mid X_{side}) = P_2 + P_4 \]
BLAST-DVC: Rate Estimation

Average of the error probabilities \( P_{n,k} \) over subimage:

\[
\overline{P}_k = \frac{1}{N} \sum_{n=1}^{N} P_{n,k}.
\]

The needed bits for the considered \( k^{th} \) bitplane can be computed as:

\[
R_k = (\overline{P}_k \cdot \log \overline{P}_k - (1 - \overline{P}_k) \cdot \log(1 - \overline{P}_k)) \cdot N + R_{CRC}
\]
BLAST-DVC: Distortion Estimation

• Estimate distortion reduction if the target bitplane is decoded.

\[ \Delta D_k = D_k - \hat{D}_k \]

- Distortion reduction
- Average distortion if the target bitplane is decoded
- Average distortion if the target bitplane is not decoded

• Average distortion estimation for a sub-image \( X_n \)

\[ D_k = \sum_{n=1}^{N} E[(X_n - \hat{X}_{n,k-1})^2]; \hat{D}_k = \sum_{n=1}^{N} E[(X_n - \text{Recon}(X_n, X'_{n,k-1}))^2] \]

- Partially reconstructed pixel value based on the previously determined \( k-1 \) bitplanes
- Partially reconstructed pixel value when the target bitplane is LDPCA-decoded \( \Rightarrow \) minimum distance symbol reconstruction is used
Minimum Distortion Reconstruction

\[ X'_{n,k} = \text{Recon}(X_n, X'_{n,k-1}) = \begin{cases} \left\lfloor \frac{X_n}{\Delta_k} \right\rfloor \cdot \Delta_k + \Delta_k - 1, & \text{if } \left\lfloor \frac{X'_{n,k-1}}{\Delta_k} \right\rfloor > \left\lfloor \frac{X_n}{\Delta_k} \right\rfloor \\ X'_{n,k-1}, & \text{if } \left\lfloor \frac{X'_{n,k-1}}{\Delta_k} \right\rfloor = \left\lfloor \frac{X_n}{\Delta_k} \right\rfloor \\ \left\lfloor \frac{X_n}{\Delta_k} \right\rfloor \cdot \Delta_k, & \text{if } \left\lfloor \frac{X'_{n,k-1}}{\Delta_k} \right\rfloor < \left\lfloor \frac{X_n}{\Delta_k} \right\rfloor \end{cases} \]
Consider that the MSB is 0 and we want to determine the next bit. 

\[ D_k = \sum_{n=1}^{N} E[(X_n - X'_{n,k-1})^2 \mid b'_{n,r} = b_{n,r}, r \in DBPs, 1 \leq r \leq k - 1] \]

\[ = \sum_{n=1}^{N} \frac{\sum_{y=L_{n,k,s}+0.5}^{S} U_{n,k,s}^{-0.5} \sum_{y=L_{n,k,s}+0.5}^{S} P(X_n = y) \cdot (y - X'_{n,k-1})^2}{\sum_{y=L_{n,k,s}+0.5}^{S} U_{n,k,s}^{-0.5} \sum_{y=L_{n,k,s}+0.5}^{S} P(X_n = y)} ; S = 2^p \text{ and } p = \text{no. of NDBPs} \]

Estimated value

Consider that the MSB is 0 and we want to determine next bit

\[ \Rightarrow \text{Next bit is 1} \]
Distortion Estimation – Bitplane LDPCA-decoded

\[
P(X | X_{side})
\]

\[
\hat{D}_k = \sum_{n=1}^{N} E[(X_n - \text{Recon}(X_n, X'_{n,k-1}))^2 | b'_{n,r} = b_{n,r}, r \in \text{DBPs}, 1 \leq r \leq k - 1]
\]

\[
= \sum_{n=1}^{N} \sum_{s=1}^{S} \sum_{y=L_{n,k,s}+0.5}^{U_{n,k,s} - 0.5} \frac{P(X_n = y) \cdot (y - \text{Recon}(y, X'_{n,k-1}))^2}{\sum_{s=1}^{S} \sum_{y=L_{i,m,n,k,s}+0.5}^{U_{i,m,n,k,s} - 0.5} P(X_n = y)}
\]

If \( y \) in Bin 00, \( \text{Recon}(y, X_{side}) \)

If \( y \) in Bin 01, \( X_{side} \) is \( \text{Recon}(y, X_{side}) \)

Copyright 2008 by L. J. Karam & W. Chien
Bitplane Decoding Selection

Once the rate $R_k$ and the distortion reduction $\Delta D_k$ are obtained, a targeted distortion-rate ratio $t$ can be chosen to determine whether bitplane decoding should be performed.

If $\Delta D_k / R_k < t$, the current bitplane is not decoded (NDBP case).

If $\Delta D_k / R_k \geq t$, CRC bits are requested followed progressively by parity/syndrome bits, one parity/syndrome bit at a time, so that error correction can be applied to the current sub-image bitplane by means of LDPCA until no errors are detected (DBP case).
**Proposed BLAST-DVC**

**Wyner-Ziv Frame Encoder**
- Divide into Sub-images
  - $X_i$
  - $X_{i,1}$
  - $X_{i,2}$
  - ... 
  - $X_{i,M}$
- Extract bitplanes
- LDPCA Encoder
- Buffer
- Block Indices Decoding

**Wyner-Ziv Frame Decoder**
- Merge Sub-images
- Minimum-distribution Pixel Reconstruction
  - $X'_i$
  - $X'_{i,1}$
  - $X'_{i,2}$
  - ... 
  - $X'_{i,M}$
- Minimum Distance Symbol Reconstruction
- LDPCA Decoder
- Block Indices Encoding

**Key frames**
- $X_{i-1}$
- $X'_{i-1}$

**Motion Compensated Interpolation**
- $\hat{X}_i$
- $\hat{X}_{i-1}$
- $\hat{X}'_{i-1}$
- $\hat{X}_{i+1}$
- $\hat{X}'_{i+1}$

**Rate-Distortion Ratio Estimation**
- $X_{i+1}$
- $X'_{i+1}$
- $X_{i+1}$
- $X'_{i+1}$
- $X_{i+1}$
- $X'_{i+1}$

**Block Indices**
- Request bits by block indices
- parity bits + CRC bits

**Copyright 2008 by L. J. Karam & W. Chien**

From Conventional to Distributed Video Coding
Minimum Distance Symbol Decoding for NDBPs

\[ P(X \mid X_{side}) \]

Assume first MSB not LDPCA-decoded and second bit is LDPCA-decoded.

\[ q_{n,k} = \arg \min_q \left\{ |\Delta_k \cdot q + \frac{\Delta_k}{2} - X_{side} - 0.5| \mid q \in Q_k \right\} \]

\( Q_k \) set containing \( S \) possible symbol values having same bits at all DBPs

If \( p \) NDBPs, \( S = 2^p \)
Outline

Distributed/Wyner-Ziv Video Coding

• Motivation
• Slepian-Wolf and Wyner-Ziv Coding
• Basic Distributed Video Coding Framework
• Existing Approaches
• Proposed BLAST-DVC
• Results

Conclusion
Results

Average PSNR versus Average Bitrate

- In most applications, the mean-squared error is used to measure quality quantitatively and is expressed in terms of a Peak-Signal-to-Noise Ratio (PSNR), which is defined in decibels (dB)

\[
PSNR = 10 \log_{10} \left( \frac{\text{peak value of reference image}}{\sigma_{mse}^2} \right)^2 = 10 \log_{10} \left( \frac{(255)^2}{\sigma_{mse}^2} \right)
\]

where

\[
\sigma_{mse}^2 = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \left| I_o(i, j) - I_p(i, j) \right|^2
\]

for a video frame of size M by N.
First 101 Frames of QCIF Mother-Daughter sequence. Key frame QP=0.
First 101 Frames of QCIF Salesman sequence. Key frame QP=0.
First 101 Frames of QCIF Foreman sequence. Key frame QP=0.
First 101 Frames of QCIF Foreman sequence. Key frame QP=0.
First 101 Frames of QCIF Mother-Daughter sequence. Key frame: QP=30 and average PSNR = 36.00 dB.
First 101 Frames of QCIF Salesman sequence. Key frame: QP=30 and average PSNR = 34.71 dB.
First 101 Frames of QCIF Foreman sequence. Key frame: QP=30 and average PSNR = 35.22 dB.
Future Research Directions

• Improved rate-distortion optimization algorithm
• Variable block size locally-adaptive DVC scheme
• Improved reconstruction scheme
• Improved DVC without feedback channel
• Real-time decoding
• Multi-View compression/3D TV
Questions ??