“COMPRESSED SENSING AND ITS APPLICATIONS IN VIDEO CODING AND CLASSIFICATION”

Panos Tsakalides
FORTH-ICS

CS-ORION: An FP7 MC-IAPP project
Outline

- Introduction
- Basics of compressive sensing (CS)
- CS in video processing for remote sensing applications
- CS in remote imaging with limited resources
- Compressive video classification
Introduction

Applications
- Remote surveillance of wide areas
- Battle damage & situation assessment
- Intelligence

Performance
- 20 – 30 hrs endurance
- TV + IR + SAR + laser designator
- Data/radio links
- Range: 200 Km (LOS), 2000 Km (satellite)
Introduction

- Increasing resolution (NTSC/PAL sensors ~20 Mbps, HD sensors ~125 Mbps)
- Available bandwidth has not increased by a similar ratio
- Reduce computational costs @ encoder to increase operational lifetime
- Exploit the increased resources of the ground control station
- Optimal video codec choice depends on our demands
Introduction

Current solutions:

- **MPEG-4**: *inter-frame* predictions
  - (-) Increased memory requirements
  - (-) Increased power consumption (motion estimation/compensation)
  - (+) Higher compression rates at lower b/w by exploiting spatio-temporal redundancies
    [e.g., HDTV signal @ 30 fps: ~5-10 Mbps]
Introduction

- Current solutions:
  - **MJPEG(2000):** (lossy) intra-frame-only video compression scheme
    - (-) Functionality tailored for static environments rather than for motion video
    - (-) Fully transmitted frame information
      [e.g., HDTV signal @ 30 fps: ~315 Mbps ]
    - (+) Low latency (typically 3 frames end-to-end)
    - (+) Low processing/memory requirements on the hardware
    - (+) Unaffected image quality @ reduced b/w (decrease fps)

  [ Bit-rate: uncompressed video > MJPEG >> MPEGx ]
Introduction

- **Motivation:** design a Compressive Video Sensing (CVS) architecture for onboard integration in video sensing devices with restricted resources
Compressed sensing

- Key assumption: *sparsity* or *compressibility* in a transform domain

CS-ORION
Compressed sensing

- Inherently wasteful process:
  - Capture all $N$ samples
  - Compute coefficient vector
  - Re-order transform coefficients
  - Thresholding

Combine sensing + compression into a single process

This is what compressive sensing (CS) does

Core concept: obtain directly a compressed set of measurements through dimensionality reduction
Compressed sensing

**CS approach:**
(signal is $K$-sparse or $K$-compressible)

@ encoder

$X \xrightarrow{\text{Compressive Sampling}}$  
$K < M \ll N$ measurements

$\xrightarrow{\text{Transmit/Store}}$

@ decoder

$M \xrightarrow{\text{Reconstruct}} N \xrightarrow{\hat{X}}$

Sensing model
$M$ linear projections

$g_i = \phi_i^T x \mid i = 1, \ldots, M \Leftrightarrow g = \Phi x$

Sensing matrix
$\Phi \in \mathbb{R}^{M \times N}$

$x = \Psi w$

$g = \Phi \Psi w$

CS-ORION
Compressed sensing

“Signal structure is local & coherent, measurements are global & incoherent”

\[ g = \Phi x = \Phi \Psi w \]

- **Random incoherent measurements:**
  - \( \Phi, \Psi \) incoherent
- **Structure & information preservation with high probability**

CS-ORION
Compressed sensing

- **Universality property:** Let $\Phi$ contain i.i.d. random entries. Then, incoherence with any fixed transform matrix is guaranteed with high probability.

- Appropriate families of matrices are the following:
  - **Gaussian matrices:** zero-mean Gaussian distribution with variance $1/N$. Exact reconstruction of $w$ (equivalently of $x$) is achieved with probability $1-O(e^{-\gamma N})$, ($\gamma>0$), if
    - $M > c K \log(N/K)$
  - **Binary matrices:** samples from the symmetric Bernoulli distribution
    - $P\{ \Phi_{mn} = +/- 1/sqrt(N) \} = \frac{1}{2}$. 
Non-linear reconstruction

- Transform-domain reconstruction

\[
\mathbf{w}^* = \arg\min_{\mathbf{w}} \| \mathbf{w} \|_1 \quad s.t. \quad \mathbf{g} = \Phi \Psi \mathbf{w}
\]

\[
\mathbf{x}^* = \Psi \mathbf{w}^*
\]

\[
\mathbf{g} = \Phi \Psi \mathbf{w} + \eta \quad \| \eta \|_2 \leq \varepsilon
\]

- Noisy measurement model

\[
\mathbf{w}^* = \arg\min_{\mathbf{w}} \| \mathbf{w} \|_1 \quad s.t. \quad \| \mathbf{g} - \Phi \Psi \mathbf{w} \|_2 \leq \varepsilon
\]

CS-ORION
Compressive Sensing – References

- Rice CS Resources:
  - http://dsp.rice.edu/cs

- Nuit Blanche blogspot:
  - http://nuit-blanche.blogspot.com/search/label/CS

- Prof. Emmanuel Candes website:

- Prof. Terence Tao website:
  - http://www.math.ucla.edu/~tao/

- Prof. David Donoho website:
  - http://www-stat.stanford.edu/~donoho/

- Our work at FORTH-ICS:
  - http://www.ics.forth.gr/~tsakalid/
CS in video processing for remote sensing applications
Exploit the efficiency of video processing standards (MPEGx) in extracting redundancies, with the power of CS in representing sparse signals.
CVS architecture

<table>
<thead>
<tr>
<th>@ Encoder</th>
<th>@ Decoder</th>
</tr>
</thead>
<tbody>
<tr>
<td>ME/MC, sparsification phase</td>
<td>Select sparsifying transform (DCT, DWT, UDWT … )</td>
</tr>
<tr>
<td>Selection of block sizes</td>
<td>Select reconstruction algorithm</td>
</tr>
<tr>
<td>(ME and CS measurement acquisition)</td>
<td></td>
</tr>
<tr>
<td>Selection of GOP size</td>
<td></td>
</tr>
<tr>
<td>Sampling operator</td>
<td></td>
</tr>
<tr>
<td>Sampling ratio (adaptive measurement allocation)</td>
<td></td>
</tr>
</tbody>
</table>

Generalized CS measurement model:

\[ g_j = \Phi(T(\Psi_c x_j)) \]

Generalized optimization problem:

\[
\min_{w_j \in \mathbb{R}^T} \| w_j \|_1 + \tau \| g_j - \Phi \Psi_c \Psi_s^{-1} w_j \|_2^2
\]
Experimental evaluation

- $\Psi_c = \{\text{DCT, DWT}\}$, $\Psi_s = \text{UDWT}$ (overcomplete)
- ME block size: 8x8
- CS block size: 32x32
- Measurement matrix: BWHT
- # quantization levels: $\{2^6, \ldots, 2^8\}$
- Sampling rate: $r = [0.05, 0.50]$ 
- GOP size: video dependent
  (Akiyo = 7, News = 6, Coastguard = 4)
Experimental evaluation

- **Quality measure:** Structural Similarity Index

\[
SSI = \frac{(2\mu_1\mu_i + c_1)(2\sigma_{II} + c_2)}{\mu_i^2 + \mu_i^2 + c_1)(\sigma_i^2 + \sigma_i^2 + c_2)}
\]

( \(\mu_i, \sigma_i\): mean, std of image I)

( \(\sigma_{II}\): correlation coefficient of original/reconstructed images)

( \(c_1, c_2\): stabilization parameters for division with a weak denominator)
Results

- General (noisy) case
Adaptive measurement allocation

- Verified intuition:
  \[ r = \frac{M_B}{n_B^2} \rightarrow \{\text{reconstruction quality & bit-rate}\} \]

- Uniform sampling for all CS blocks

- Adaptive measurement allocation

Residual frame

\[
\hat{\sigma}_n = \frac{\text{MAD}}{n_B} \quad \text{(MAD estimate of noise std)}
\]

\[
\hat{\sigma}_n = \sigma_n \sqrt{\frac{2 \log(N^2)}{N}}
\]

\[
r_j = \frac{1}{n_B^2} \cdot \min(\text{card}\{C_j > \varrho_{Th}\}, K_{max})
\]

\[
K_{max} = r \cdot n_B^2
\]
Results

CS-ORION
Conclusions

CVS vs. MPEG-2:

- (+) Comparable performance with MPEG-2 @ lower bit-rates, especially for rapidly varying content
- (+) Increased robustness @ low input SNR
- (-) Increased computational cost at encoder (as in MPEG-2)
Future work

- Decrease computational complexity at encoder by transferring ME/MC at the decoder
- Optimal (and automatic) way to specify system parameters (GOP size, sampling operator, regularization parameters) to adapt to the frame statistics
Introduction

- **Motivation:** in a lightweight imaging system motion estimation @ encoder should be avoided
- **Separate encoding – Separate/joint decoding**
  - **Main drawbacks:**
    1. Spatio-temporal redundancies are not removed @ encoder (increased bit-rates)
    2. Sensitive to propagation of reconstruction errors
- **Efficiency of MPEGx family is due to intra-frame transform coding and inter-frame motion prediction**
- **Encoder with increased memory & processing resources is required**
Encoder

Video sequence → I-frame

8x8 blocks

DCT → Quantization → Coding

(JPEG)

Lossless Huffman coding (with recursive splitting)

$D_q(m,n) = \text{round}\left(\frac{D(m,n)}{S \cdot Q(m,n)}\right)$

$Q = \begin{bmatrix}
8 & 16 & 19 & 22 & 26 & 27 & 29 & 34 \\
16 & 16 & 22 & 24 & 27 & 29 & 34 & 37 \\
19 & 22 & 26 & 27 & 29 & 34 & 34 & 38 \\
22 & 22 & 26 & 27 & 29 & 34 & 37 & 40 \\
22 & 26 & 27 & 29 & 32 & 35 & 40 & 48 \\
26 & 27 & 29 & 32 & 35 & 40 & 48 & 58 \\
26 & 27 & 29 & 34 & 38 & 46 & 56 & 69 \\
27 & 29 & 35 & 38 & 46 & 56 & 69 & 83 \\
\end{bmatrix}$

CS-ORION
Lossless Huffman coding (with recursive splitting)

same measurement matrix for all blocks
(BWHT operator is used due to “hardware-friendly” implementation)

Uniform scalar quantization

$g_j = \Phi_j x_j$, $j = 1, \ldots, n_B$

j-th block (space-domain)

# blocks
Decoder

(inverse JPEG)

CS-ORION
Decoder

Termination criteria:
• Max number of iterations $L_{max}$
• Reconstruction error $||\hat{x}_j^{n+1} - x_j^n||_2 \leq \varepsilon$

Threshold specification: $\rho_{Th} = \lambda \sigma \sqrt{2 \log(B^2)}$

$$\sigma = \frac{\text{median}(|\hat{w}_j^{n+1}|)}{0.6745}$$

$$\hat{w}_j^{n+1} = \Psi_s(\hat{x}_j^{n+1})$$

Hard thresholding operator
Sparsifying transform

(IHT)

$$\hat{x}_j^{n+1} = \hat{x}_j^n + \Phi^T (g_j - \Phi \hat{x}_j^n)$$
$$\hat{x}_j^{n+1} = \Psi_s^{-1}(T \{\hat{\Psi}_s(\hat{x}_j^{n+1})\})$$

CS-ORION
Decoder

- Increased robustness when working directly with the (equally important) CS measurements
- Sub-pixel motion estimation (increased resources at decoder)

\[
\begin{align*}
\hat{R} &= \hat{P} - \mathcal{D}\{\hat{I}_{MC}\} \\
\Phi\hat{R} &= \Phi(\hat{P} - \mathcal{D}\{\hat{I}_{MC}\}) \\
\Phi\hat{x} &= \Phi\hat{P} - \Phi\mathcal{D}\{\hat{I}_{MC}\} \\
g_{error} &= g - g_{MC} \\
g_{error} &\xrightarrow{IHT} \hat{R} \\
\hat{P}_{n+1} &= \mathcal{D}\{\hat{I}_{MC}\} + \hat{R} \\
\end{align*}
\]
Decoder

Super-resolution
- Superior than usual 2-D interpolation
- Coupled trained dictionaries: $D_{HR}$ – high-resolution patches, $D_{LR}$ – low-resolution patches

Use sparse representation in $D_{LR}$ to reconstruct the corresponding high-res patch from $D_{HR}$

1. Initial training with arbitrary images
2. Update by incorporating reconstructed I-frame patches

* H. Zhang et al., “Efficient sparse representation based image super resolution via dual dictionary learning”, (ICME’11)

CS-ORION
Experimental evaluation

- **Setup @ encoder**
  - GOP size: 6
  - Block size: 16x16
  - Measurement matrix: BWHT
  - # quantization levels: \( \{2^6, \ldots, 2^8\} \)
  - Sampling rate: \( r = 0.10 \) (\( M = 26 \) CS measurements/block)
  - Downsampling factor: 2

- **Setup @ decoder**
  - \( \lambda = 3, \ L_{\text{max}} = 400, \ e = 10^{-4} \)
  - \( C_{\text{max}} = 10 \)
Experimental evaluation
Conclusions

CVS vs. MJPEG:

- (+) Straightforward embedding of CS in standard MJPEG, no additional cost at the encoder
- (+) CS + refining ME/MC at the decoder outperforms MJPEG (at similar bit-rates)
  (refinement @ MJPEG decoder is impossible)
- (-) Iterative reconstruction and dictionary updating need careful handling to reduce latency
Future work

- **@ encoder:**
  - CS-only encoding using single-pixel camera (lower acquisition expense at wavelengths were standard cameras are "costly")

- **@ decoder**
  - Improve quality by improving the initial reconstruction of P-frames
  - Fast updating of dictionary for real-time super-resolution (in systems with time limitations)

- Joint compressive super-resolution & refinement to avoid the "wavy" motion
Compressive video classification


CS-ORION
Introduction

- Conventional approaches require full-res video data for the extraction of descriptors (color histograms, optical flow vectors, shape)
- Typical classification techniques (SVM, HMM, MAP)
- Onboard processing is prohibitive in case of limited power/memory resources, base-station processing may be prohibitive in case of limited bandwidth
- Motivation: video classification in a decision system with limited resources without handling original high-res data
- Exploit the properties of linear random projections
Introduction

- Assumption: system equipped with a single-pixel camera

- 2-phase process:
  - Feature extraction (generate a compact representation in a low-dimensional space)
  - Classification (similarity measurement using a supervised learning approach)
Video classification system
Feature extraction

- video sequence with R frames: \( V = \{x_1, \ldots, x_R\} \)

- CS domain representation: \( x_j \xrightarrow{\Phi} G_j = \{g^j_1, \ldots, g^j_s\} \) \( g^j_i = \Phi p^j_i \)

\( (\text{block-wise}) \)

\( V \xrightarrow{\Phi} G = \{G^1, \ldots, G^R\} \)

- Training set: \( G_k = [g^1_{1k}, \ldots, g^1_{sk}, \ldots, g^R_{1k}, \ldots, g^R_{sk}] \) \( k = 1, \ldots, CQ \)
Classification

- Approach 1: exploit directly the CS feature vectors
  - Nearest-neighbor rule: 
    \[ c^* = \arg \min_{c \in \{1, \ldots, C\}, k = 1, \ldots, CQ} \| z_{query} - z_k \|_2^2 \]
  - Multi-class SVM (with a 1-against-1 approach): one SVM for each pair of classes

\[ d_{ij}(y) \quad \text{- Discriminant function for classes } (i, j) \]

\[ d_{ij}(f_{CS,\text{query}}) \begin{cases} > 0 & \text{, a vote is assigned to the i-th class} \\ \text{else} & \text{, a vote is assigned to the j-th class} \end{cases} \]

Select class with the max number of votes

CS-ORION
Classification

- Approach 2: solve a convex optimization problem to recover a sparse class-indicator vector

\[ \alpha = [\alpha_1^1, \ldots, \alpha_Q^1, \ldots, \alpha_1^i, \ldots, \alpha_Q^i, \ldots, \alpha_1^C, \ldots, \alpha_Q^C] \in \mathbb{R}^{CQ} \]

(for the i-th class)

\[ \alpha = [0, \ldots, 0, \alpha_1^i, \ldots, \alpha_Q^i, 0, \ldots, 0] \]

- Solve a convex problem to recover sparse support: (we used OMP)

\[ \alpha^* = \arg \min_{\alpha \in \mathbb{R}^{CQ}} \|\alpha\|_1 \text{, s.t. } \|z_{query} - D\alpha\|_2 < \epsilon \]
Results

- **Dataset:**
  - UCF50 database, 8 classes of activities
  - Challenging due to variations in camera motion, object appearance/pose, illumination conditions
  - 50 videos of 50 frames per class
  - Block-size: 32 x 32
  - 50 Monte-Carlo runs, different separation in K training and 50-K testing samples (K = {12, 24, 36})
  - Block Walsh-Hadamard measurement matrix
  - Sampling ratio (M/N) varies in [0.01, 0.20]
Results

![Graphs showing mean success rate vs. sampling rate for 12 and 36 training samples.]

\[
\text{success rate} = \frac{\text{number of correctly classified sequences}}{\text{total number of query sequences}}
\]
Conclusions

- Incoherent random projections were shown to be representative of the inherent video content.
- Increased classification accuracy without accessing high-res video data.
- SVM classification was shown to be more robust for CS-based feature vectors.

CS-ORION
Future work

- Frame sparsity is not exploited, introduce an intermediate linear dimensionality reduction step (e.g., PCA, embedding in a low-dimensional manifold)
- Increase classification margin by exploiting color information to generate CS features
Thank you!