"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

Patroller™R

Applications

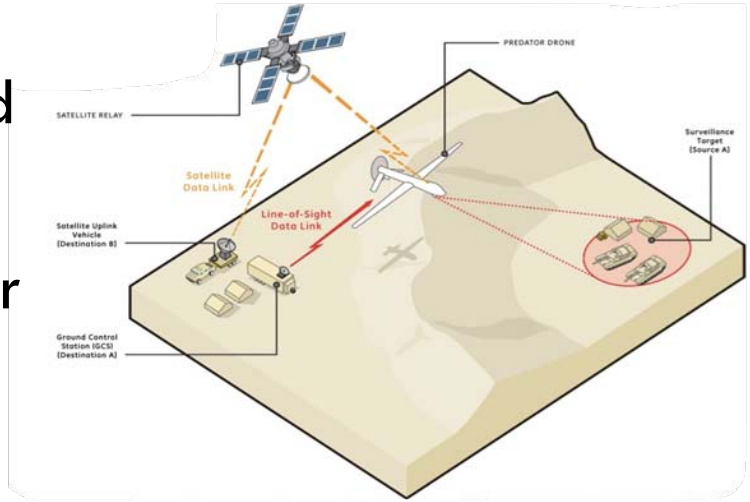
- Remote surveillance of wide areas
- Battle damage & situation assessment
- Intelligence

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

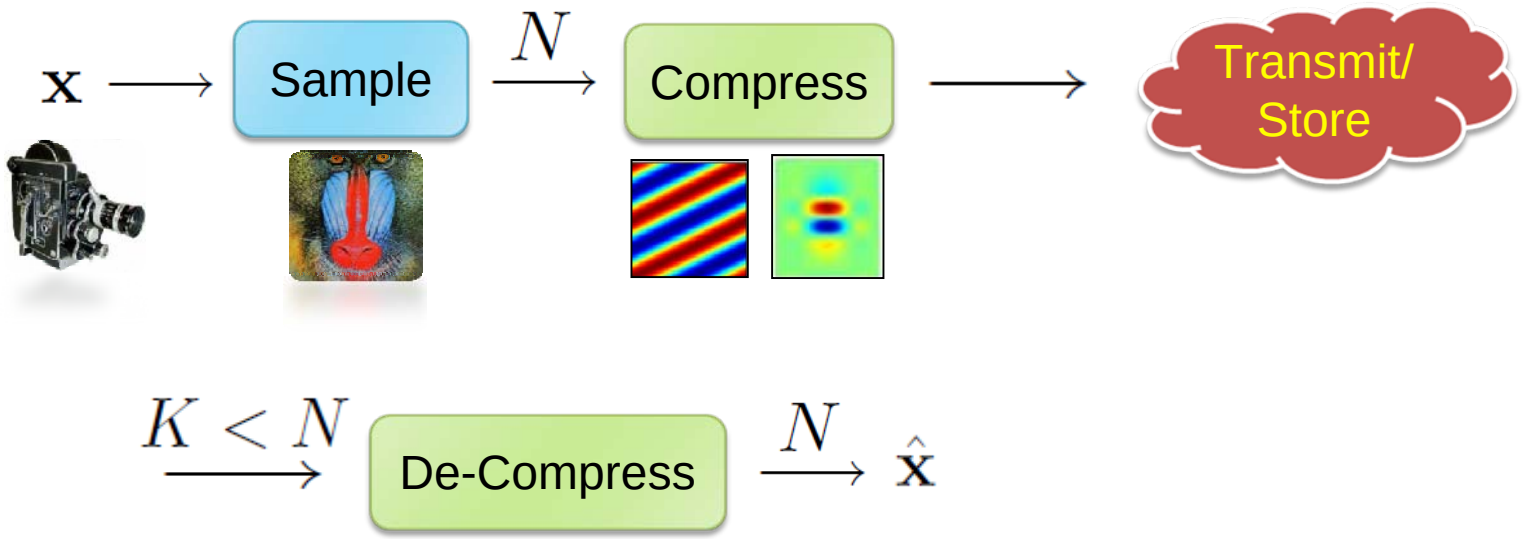
- Current solutions:
  - MJPEG(2000): (lossy) *intra-frame-only* video compression scheme
    - (-) Functionality tailored for static environments rather than for motion video
    - (-) Fully transmitted frame information
      [e.g., HDTV signal @ 30 fps: ~315 Mbps]
    - (+) Low latency (typically 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

Compressed sensing

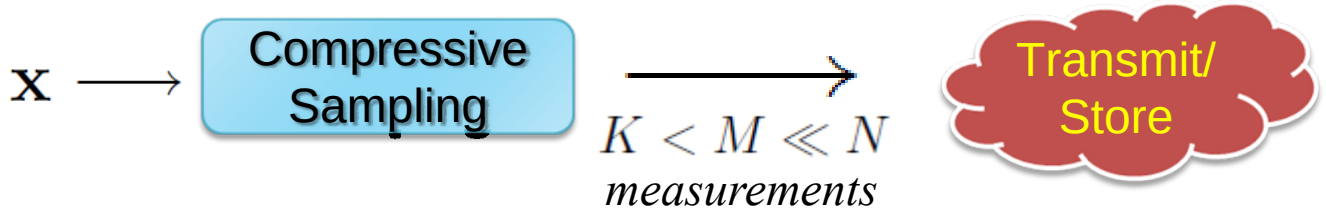
- (-) Inherently wasteful process:
  - Capture all *N* samples
  - Compute coefficient vector
  - Re-order transform coefficients
  - Thresholding
- Combine sensing + compression into a single process
- This is what *compressive sensing* (CS) does
- Core concept: obtain directly a compressed set of *measurements* through *dimensionality reduction*

Compressed sensing

CS approach:

(signal is K-sparse or K-compressible)

@ encoder



@ decoder

Sensing matrix

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

Sensing model

M *linear* projections

$$g_i = \phi_i^T \mathbf{x}\,|_{i=1,\ldots,M} \Leftrightarrow \mathbf{g} = \Phi\mathbf{x}$$

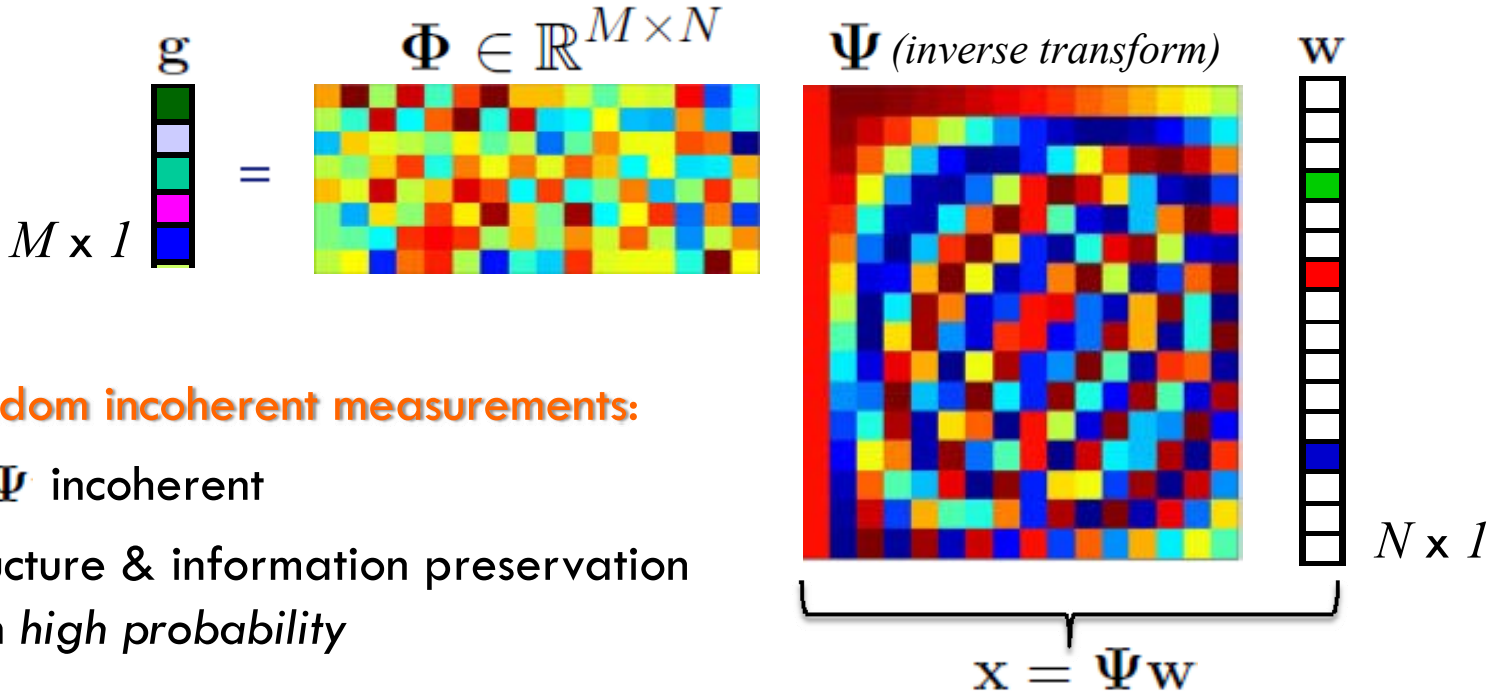
$$\mathbf{x} = \Psi\mathbf{w}$$

$$\mathbf{g} = \Phi\Psi\mathbf{w}$$

Compressed sensing

" Signal structure is local & coherent, measurements are global & incoherent"

$$\mathbf{g} = \mathbf{\Phi x} = \mathbf{\Phi \Psi w}$$



- **Random incoherent measurements:**

  $\mathbf{\Phi}$, $\mathbf{\Psi}$ incoherent

- Structure & information preservation with *high probability*

Compressed sensing

- **Universality property:** Let $\boldsymbol{\Phi}$ 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 $\boldsymbol{w}$ (equivalently of $\boldsymbol{x}$) 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
  - $P\{ \boldsymbol{\Phi}_{mn} = +/- 1/\sqrt{N} \} = ½.$

Non-linear reconstruction

- Transform-domain reconstruction

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

$$\mathbf{x}^* = \mathbf{\Psi w}^*$$

$$\mathbf{g} = \mathbf{\Phi\Psi w} + \eta \rightarrow \|\eta\|_2 \leq \varepsilon$$

- Noisy measurement model

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

Compressive Sensing – References

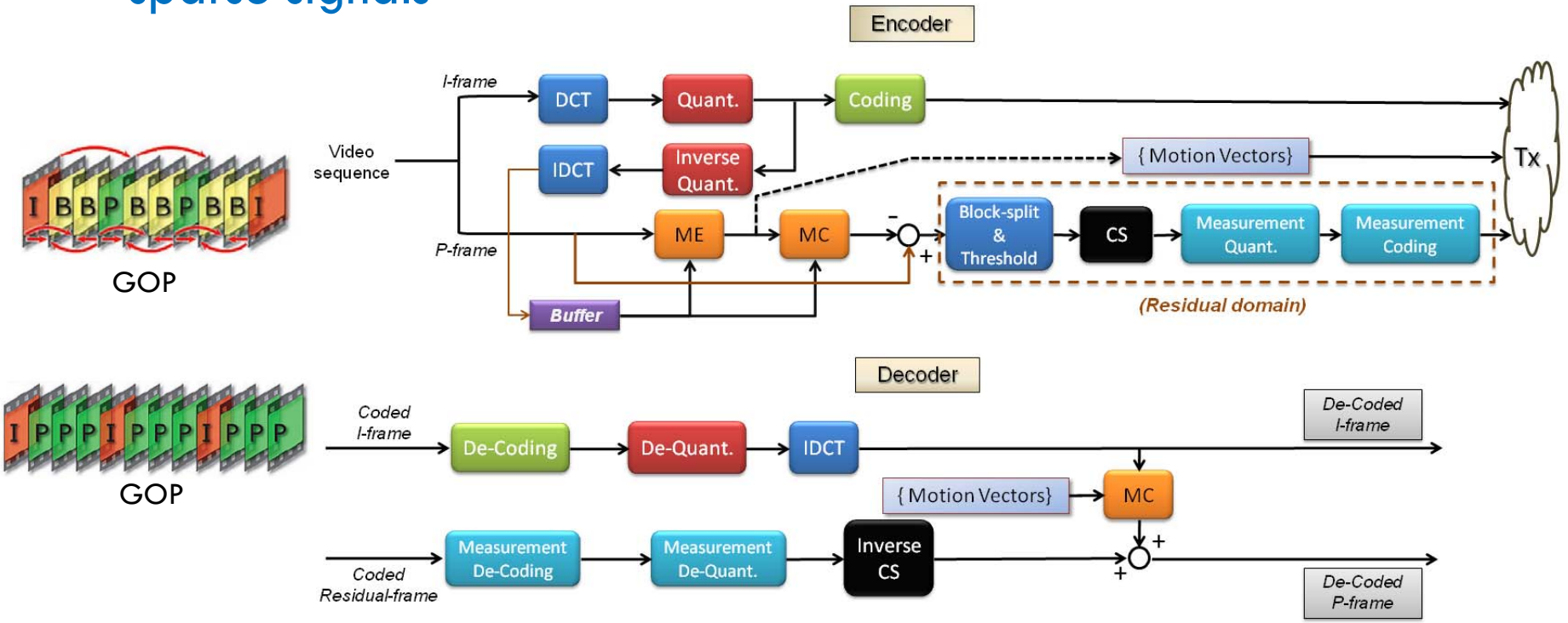
- Rice CS Resources:
  - http://dsp.rice.edu/cs
- Nuit Blanche blogspot:
  - http://nuit-blanche.blogspot.com/search/label/CS
- Prof. Emmanuel Candes website:
  - http://www-stat.stanford.edu/~candes/
- Prof. Terence Tao website:
  - http://www.math.ucla.edu/~tao/
- Prof. David Donoho website:
  - http://www-stat.stanford.edu/~donoho/

- Our work at FORTH-ICS:
  - http://www.ics.forth.gr/~tsakalid/

CS in video processing for remote sensing applications

CVS architecture

- Exploit the efficiency of video processing standards (MPEGx) in extracting redundancies, with the power of CS in representing sparse signals



CVS architecture

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

Generalized CS measurement model:

$$g_j = \Phi(\mathcal{T}(\Psi_c x_j))$$

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

- Ψ_c = {DCT, DWT} , Ψ_s = 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

$$\mathrm{SSI} = \frac{(2\mu_{\mathrm{I}}\mu_{\hat{\mathrm{I}}} + c_1)(2\sigma_{\mathrm{I}\hat{\mathrm{I}}} + c_2)}{(\mu_{\mathrm{I}}^2 + \mu_{\hat{\mathrm{I}}}^2 + c_1)(\sigma_{\mathrm{I}}^2 + \sigma_{\hat{\mathrm{I}}}^2 + c_2)}$$

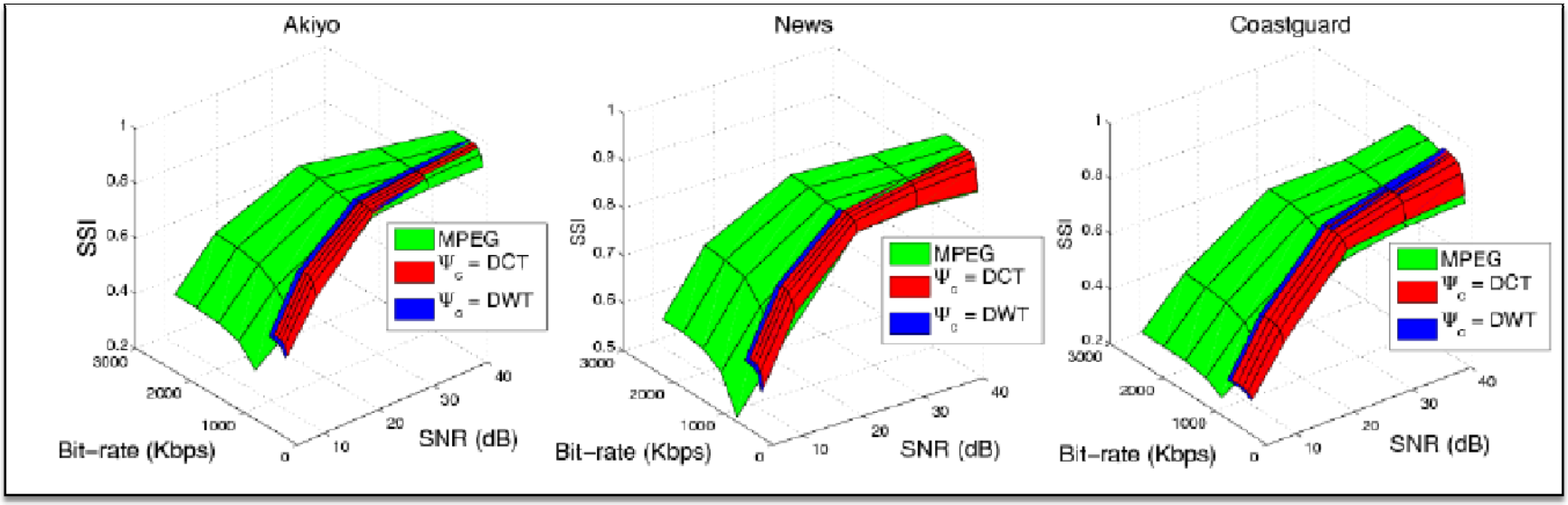
(μ_I , σ_I : mean , std of image I)

($\sigma_{\mathrm{I}\hat{\mathrm{I}}}$ 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:

  $r = \frac{M_B}{n_B^2}$ ↗ ➡ {reconstruction quality & bit-rate} ↗

- Uniform sampling for all CS blocks
- *Adaptive measurement allocation*

Residual frame

MAD estimate of noise std

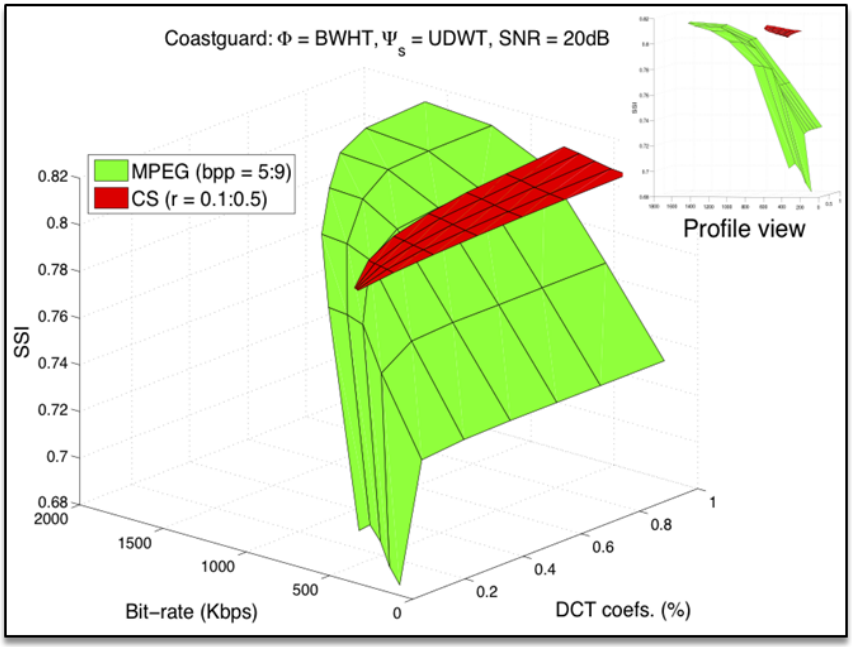
Block DCT thresholding

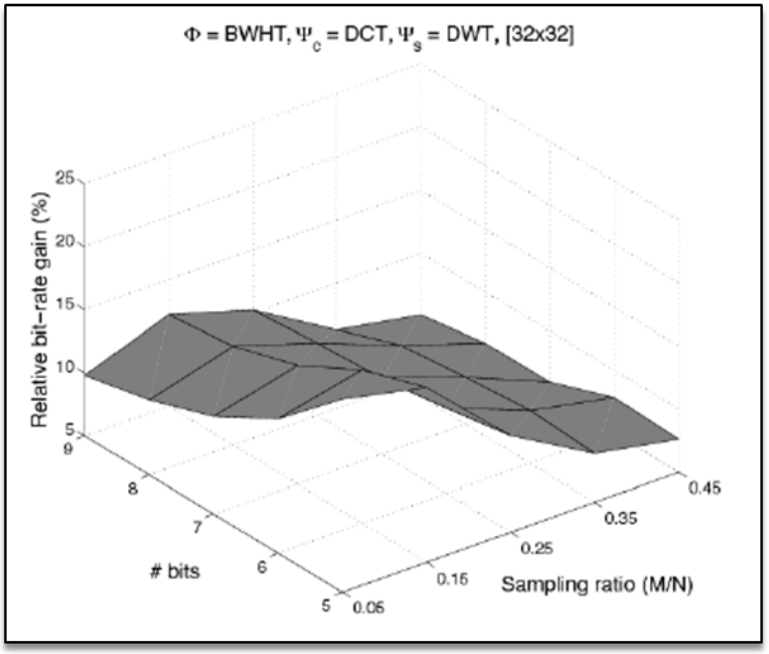
$\varrho_{Th} = \lambda \sigma_\eta \sqrt{2 \log(N^2)}$

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

$K_{max} = r \cdot n_B^2$

Results


Coastguard: Φ = BWHT, Ψs = UDWT, SNR = 20dB
MPEG (bpp = 5:9)
CS (r = 0.1:0.5)
Profile view
SSI
0.82
0.8
0.78
0.76
0.74
0.72
0.7
0.68
2000
1500
1000
500
0
Bit−rate (Kbps)
0.2
0.4
0.6
0.8
1
DCT coefs. (%)


Φ = BWHT, Ψc = DCT, Ψs = DWT, [32x32]
Relative bit−rate gain (%)
25
20
15
10
5
9
8
7
6
5
bits
0.05
0.15
0.25
0.35
0.45
Sampling ratio (M/N)

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

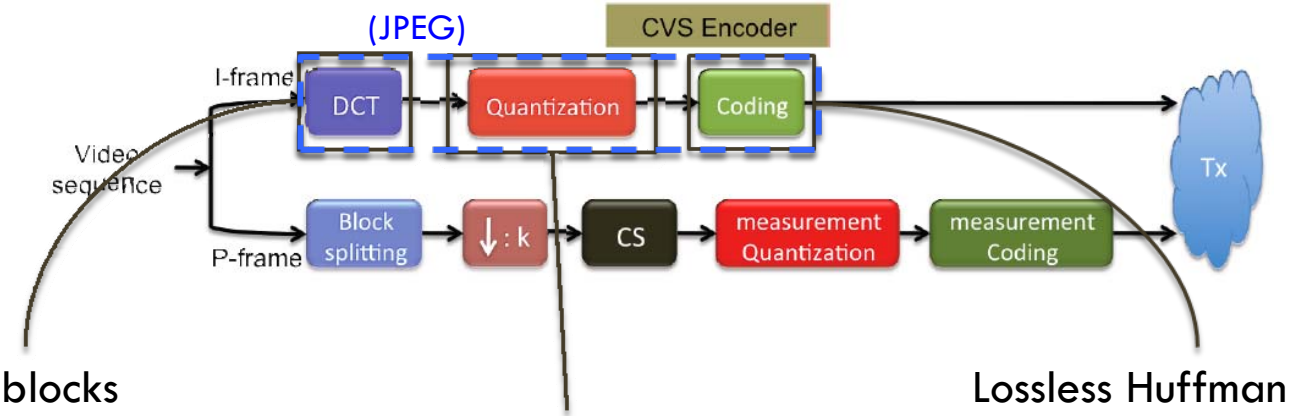
CS in remote imaging with limited resources

G. Tzagkarakis, A. Woiselle, P. Tsakalides, and J. L. Starck, "Design of a Compressive Remote Imaging System Compensating a Highly Lightweight Encoding with a Refined Decoding Scheme," in *Proc. International Conference on Computer Vision Theory and Applications (VISAPP '12)*, Rome, Italy, February 24-26, 2012.

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



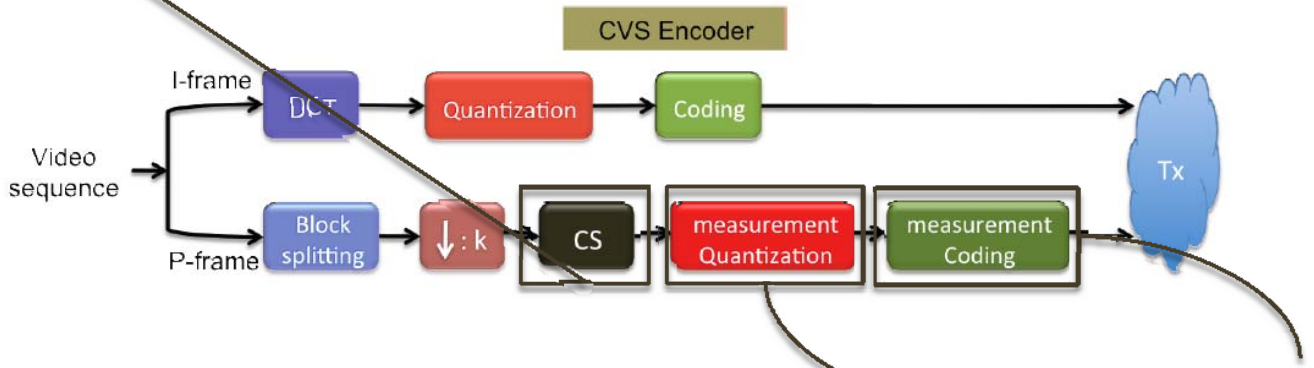
8x8 blocks

Lossless Huffman coding (with recursive splitting)

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

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

Encoder



$$\mathbf{g}_j = \Phi_j \mathbf{x}_j, \quad j = 1, \ldots, n_B$$

Lossless Huffman coding (with recursive splitting)

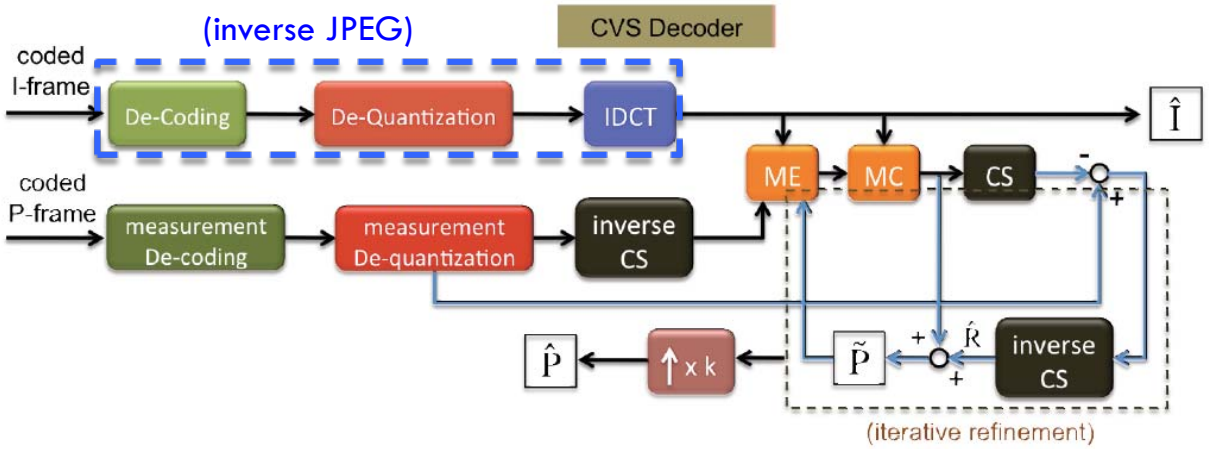
same measurement matrix for all blocks (BWHT operator is used due to "hardware-friendly" implementation)

j-th block (**space-domain**)

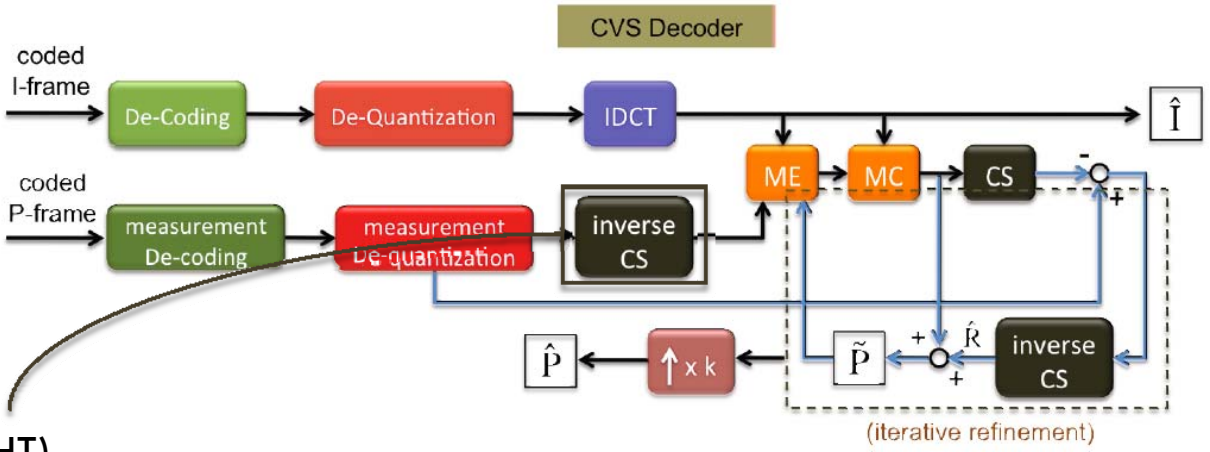
blocks

Uniform scalar quantization

Decoder


(inverse JPEG)
CVS Decoder
coded
I-frame
De-Coding
De-Quantization
IDCT
Î
coded
P-frame
measurement
De-coding
measurement
De-quantization
inverse
CS
ME
MC
CS
-
+
P̂
x k
P̃
+
+
Ŕ
inverse
CS
(iterative refinement)

Decoder



(IHT)

$$\tilde{\mathbf{x}}_j^{n+1} = \hat{\mathbf{x}}_j^n + \Phi^T(\mathbf{g}_j - \Phi\hat{\mathbf{x}}_j^n)$$

$$\hat{\mathbf{x}}_j^{n+1} = \Psi_s^{-1}\left(\mathcal{T}\{\Psi_s(\tilde{\mathbf{x}}_j^{n+1})\}\right)$$

Hard thresholding operator

Sparsifying transform

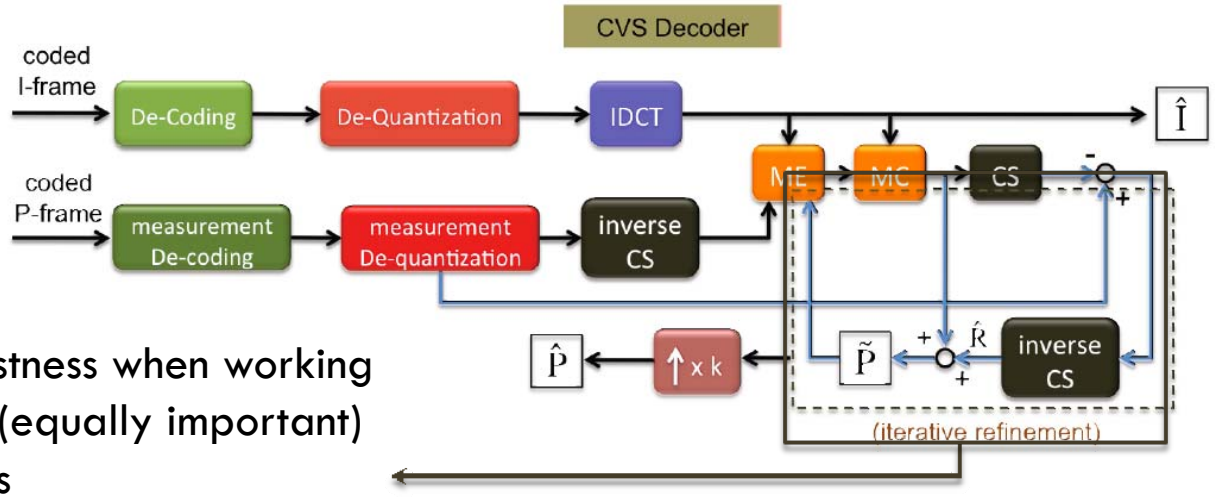
Termination criteria:

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

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

$$\sigma = \frac{\text{median}(|\tilde{\mathbf{w}}_j^{n+1}|)}{0.6745} \qquad \tilde{\mathbf{w}}_j^{n+1} = \Psi_s(\tilde{\mathbf{x}}_j^{n+1})$$

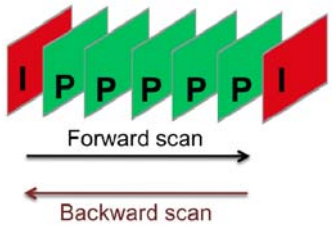
Decoder



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

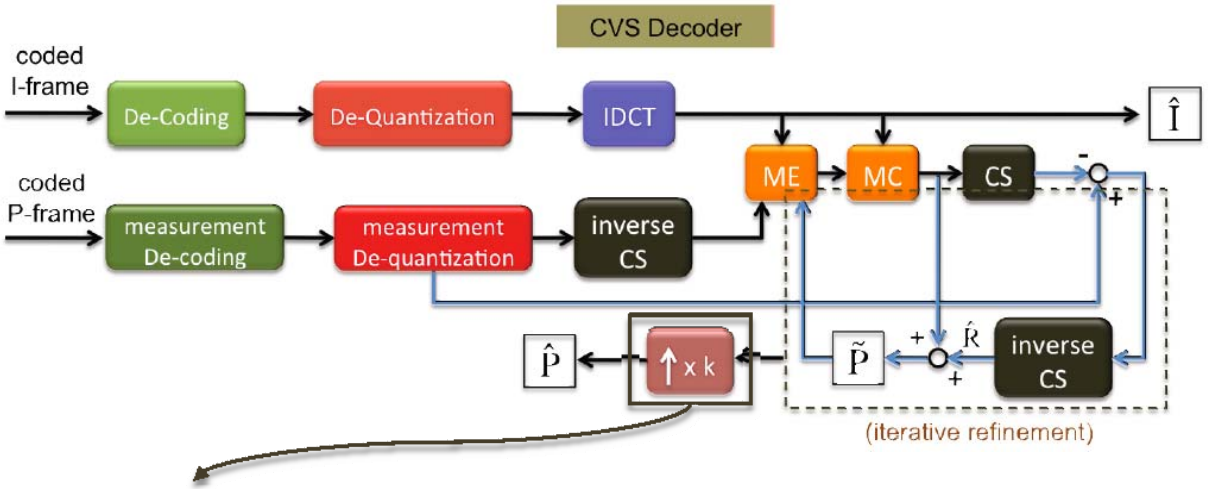
$$
\begin{aligned}
\hat{\mathrm{R}} &= \hat{\mathrm{P}} - \mathcal{D}\{\hat{\mathrm{I}}_{MC}\} \Rightarrow \\
\Phi\hat{\mathrm{R}} &= \Phi(\hat{\mathrm{P}} - \mathcal{D}\{\hat{\mathrm{I}}_{MC}\}) \Rightarrow \\
\Phi\hat{\mathrm{R}} &= \Phi\hat{\mathrm{P}} - \Phi\mathcal{D}\{\hat{\mathrm{I}}_{MC}\} \overset{\mathbf{g}_j = \Phi\mathcal{D}\{\mathbf{x}_j\}}{\Rightarrow} \\
\mathbf{g}_{error} &= \mathbf{g} - \mathbf{g}_{MC} \,.
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{g}^n_{MC} &= \Phi\mathcal{D}\{\hat{\mathrm{I}}^n_{MC}\} \\
\mathbf{g}^n_{error} &= \mathbf{g} - \mathbf{g}^n_{MC} \\
\mathbf{g}^n_{error} &\xrightarrow{\text{IHT}} \hat{\mathrm{R}}^n \\
\hat{\mathrm{P}}^{n+1} &= \mathcal{D}\{\hat{\mathrm{I}}^n_{MC}\} + \hat{\mathrm{R}}^n
\end{aligned}
$$



CS-ORION

Decoder



Super-resolution

- Superior than usual 2-D interpolation
- Coupled trained dictionaries:

$\mathbf{D}_{HR}$ – high-resolution patches
$\mathbf{D}_{LR}$ – low-resolution patches

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

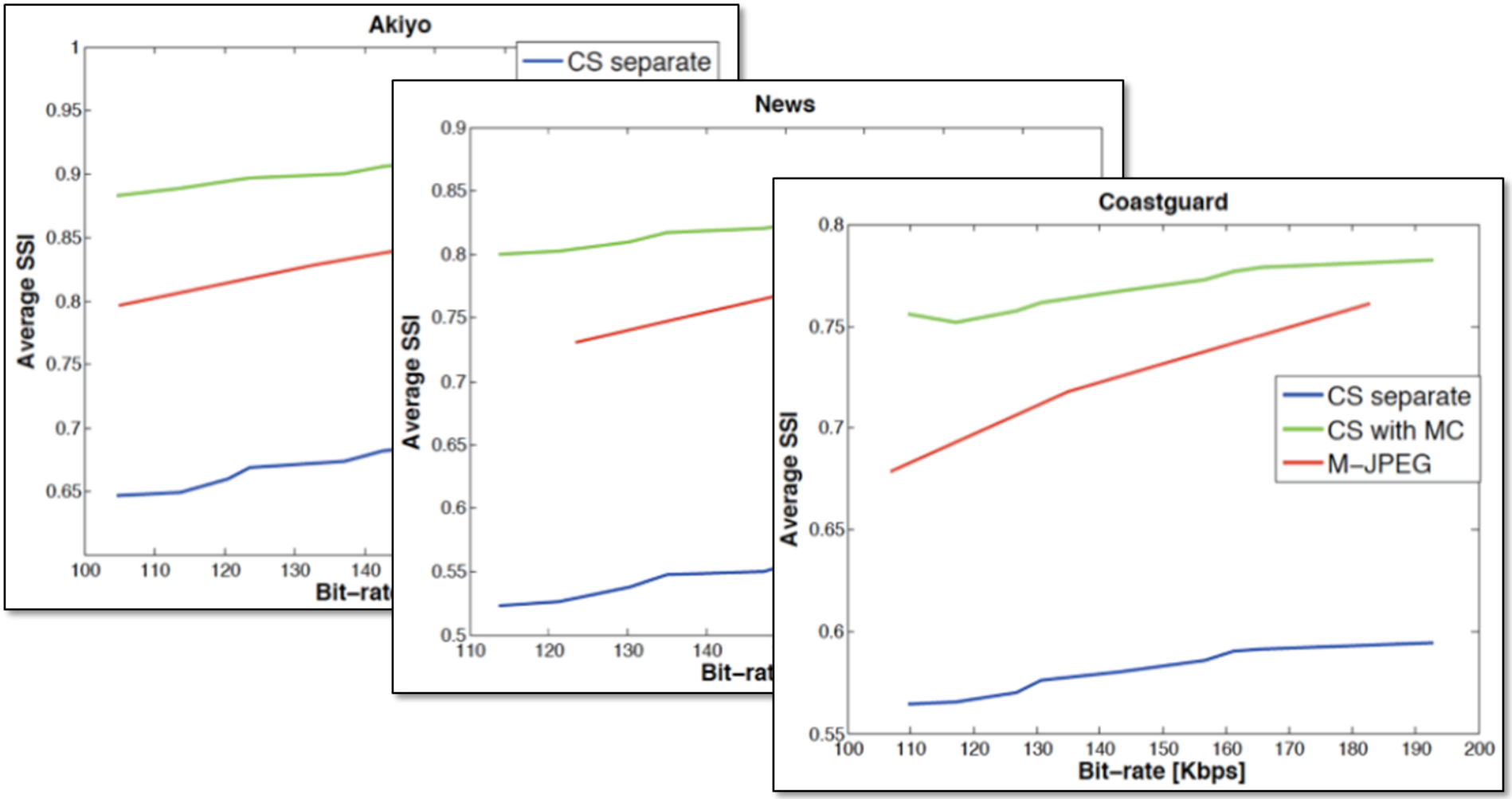
Use sparse representation in $\mathbf{D}_{LR}$ to reconstruct the corresponding high-res patch from $\mathbf{D}_{HR}$

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

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
  - $\lambda = 3,\ L_{max} = 400,\ \varepsilon = 10^{-4}$
  - $C_{max} = 10$

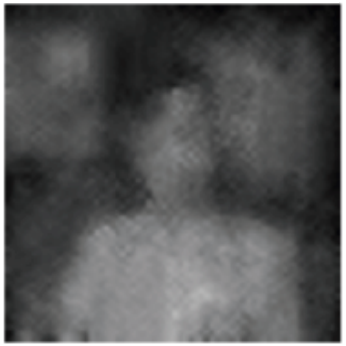
Experimental evaluation



Original

M-JPEG

CS w/o MC



CS with MC

Original

M-JPEG

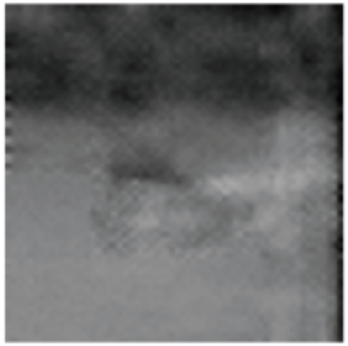
CS w/o MC

CS with MC

Original

M-JPEG

CS w/o MC



CS with MC

CS-ORION

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

G. Tzagkarakis, P. Charalampidis, G. Tsagkatakis, J. L. Starck, and P. Tsakalides, "Compressive Video Classification for Decision Systems with Limited Resources," *in Proc. Picture Coding Symposium (PCS'12)*, Krakow, Poland, May 7-9, 2012.

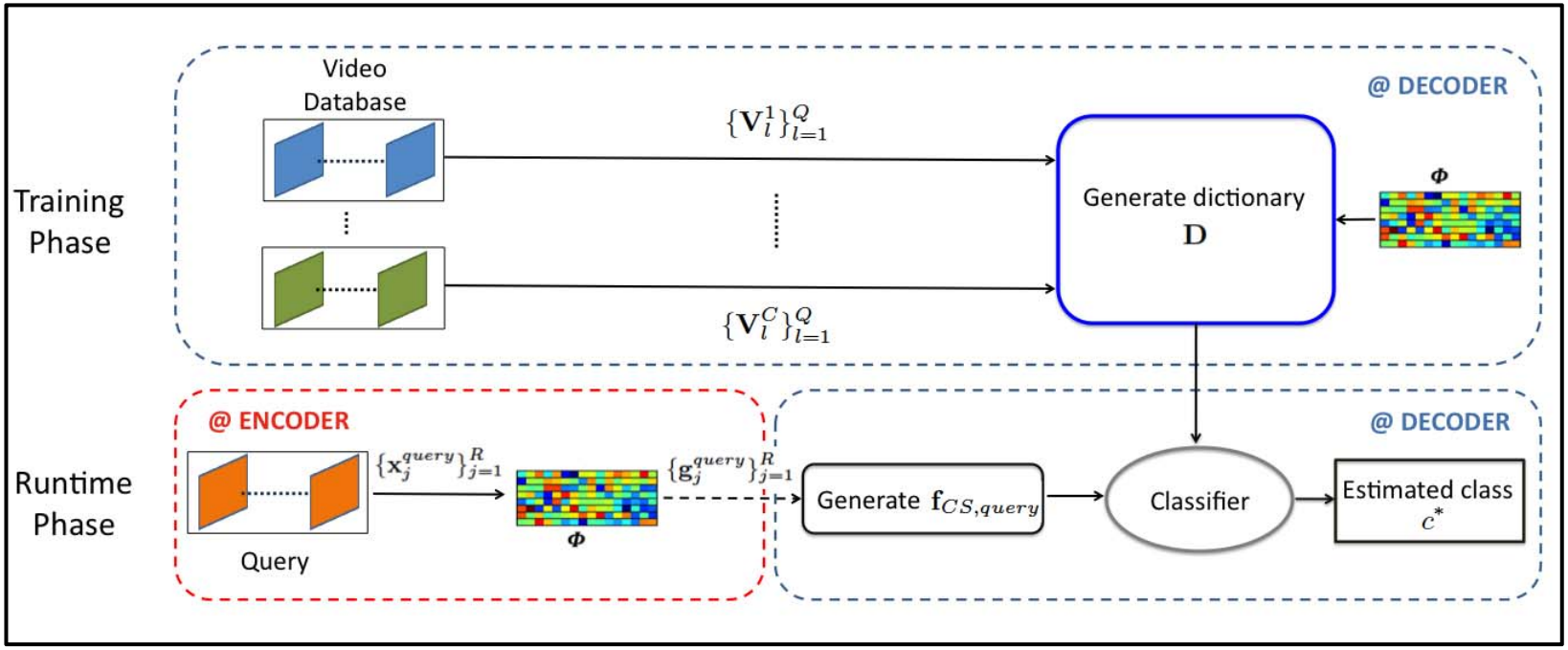
Introduction

- 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: $\mathbf{V} = \{\mathbf{x}_1, \ldots, \mathbf{x}_R\}$
- CS domain representation: $x_j \xrightarrow{\Phi} G^j = \left\{ g_1^j, \ldots, g_S^j \right\} \quad g_i^j = \Phi p_i^j$

  (block-wise)

  $\mathbf{V} \xrightarrow{\Phi} G = \left\{ G^1, \ldots, G^R \right\}$

- Training set: $G_k = \left[g_{1k}^1, \ldots, g_{Sk}^1, \cdots, g_{1k}^R, \ldots, g_{Sk}^R \right] \quad k = 1, \ldots, CQ$

Classification

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

$d_{ij}(\mathbf{y})$ - Discriminant function for classes (*i*, *j*)

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

Select class with the max number of votes

Classification

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

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

(for the i-th class)

$$\boldsymbol{\alpha} = [0, \ldots, 0, \alpha_1^i, \ldots, \alpha_Q^i, 0, \ldots, 0]$$

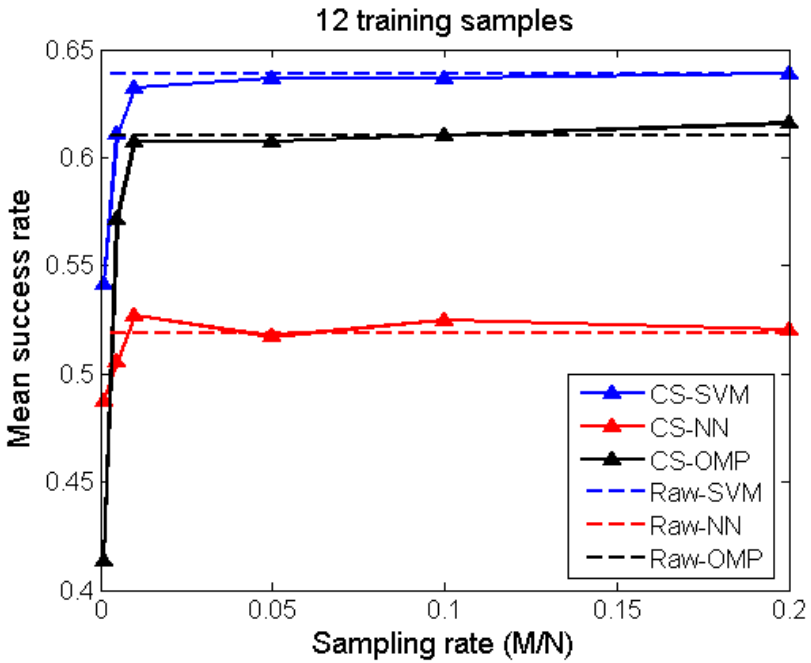
- Solve a convex problem to recover sparse support: (we used OMP)

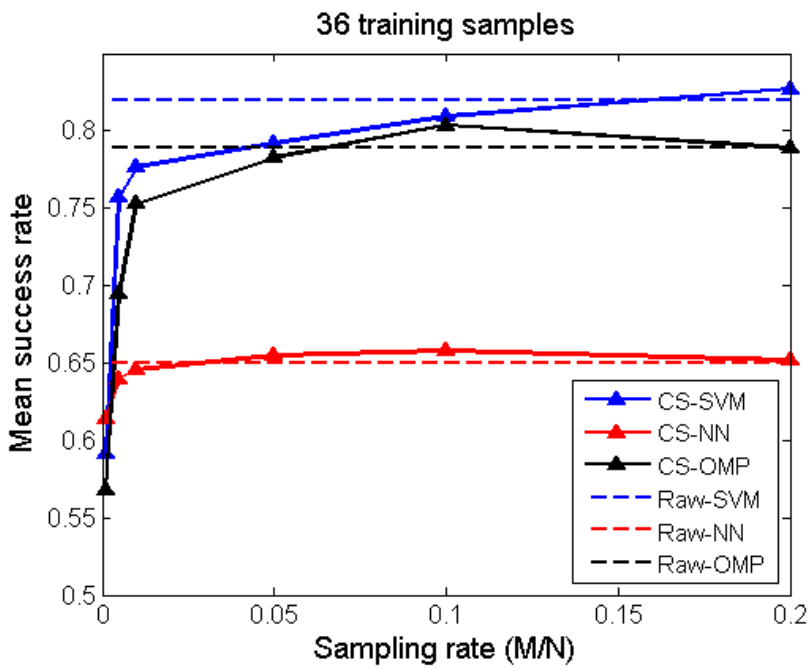
$$\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha} \in \mathbb{R}^{CQ}} \|\boldsymbol{\alpha}\|_1 \ , \ \text{s.t.} \ \|z_{query} - \mathbf{D}\boldsymbol{\alpha}\|_2 < \epsilon$$

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





$$success\ rate = \frac{number\ of\ correctly\ classified\ sequences}{total\ number\ of\ query\ sequences}$$

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

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!