Block-Based Compressed Sensing of Images and Video

James E. Fowler

Department of Electrical & Computer Engineering
Geosystems Research Institute
Mississippi State University

March 2010

MISSISSIPPI STATE UNIVERSITY™

Outline

Compressed Sensing (CS)

What is CS?

Emerging mathematical paradigm permitting:

- Sampling at sub-Nyquist rates via linear projection onto a measurement basis of lower dimension
- Exact reconstruction when signal is sparse in some transform domain
- Approximate reconstruction when signal is compressible in some transform domain
- Random measurement matrix works universally for all signals with high probability
- Also know as: compressive sensing, compressive sampling

CS Overview

Goal

Recover vector $\mathbf{x} \in \Re^N$ from

$$\mathbf{y} = \mathbf{\Phi}\mathbf{x} \in \Re^M$$

- $\mathbf{\Phi}$: $M \times N$ measurement matrix, $M \ll N$
- Usually, $\mathbf{\Phi}$ is a random matrix
- Subsampling rate, or subrate, is M/N

The measurement process $\mathbf{\Phi}\mathbf{x}$ is accomplished within sensing device:

- $\mathbf{x}$ is acquired and simultaneously reduced in dimension

CS Overview

Fundamental Tenet of CS

Recovery is exact if $\mathbf{x}$ is sufficiently sparse:

- L-sparsity: only L coefficients of

$$\check{\mathbf{x}} = \boldsymbol{\Psi}\mathbf{x}$$

are nonzero for some transform $\boldsymbol{\Psi}$

Approximate Recovery

Real-world signals—often not sparse but compressible:

$$|\check{x}_n| < Cn^{-1/p}$$

where $p \leq 1$, $C < \infty$, and $\check{x}_n$ are sorted coefficients of $\check{\mathbf{x}}$

- Recovery is close to L-sparse approximation to $\check{\mathbf{x}}$

CS Overview

Ideal Recovery: ℓ_0

- Find $\check{\mathbf{x}}$ with smallest ℓ_0 norm consistent with $\mathbf{y}$:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{\Psi x}\|_0 \qquad \text{s.t.} \qquad \mathbf{y} = \mathbf{\Phi x}$$

- Computationally infeasible for all but the smallest of problems

CS Overview

Practical Recovery: Basis Pursuit (BP)

Convex relaxation of ℓ_0 problem:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\Psi\mathbf{x}\|_1 \qquad \text{s.t.} \qquad \mathbf{y} = \Phi\mathbf{x}$$

- Implemented via linear programming
- High computational complexity in practice
- Relaxed/greedy variants of BP, e.g.:
  - gradient projection sparse reconstruction (GPSR)
  - sparsity adaptive matching pursuits (SAMP)

Project Landweber (PL) Recovery

Approximate Recovery

- For compressible signals, relax equality constraint and replace constrained ℓ_1 recovery with unconstrained optimization:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{\Psi x}\|_1 + \lambda \|\mathbf{y} - \mathbf{\Phi x}\|_2$$

- Popular solution: iterative thresholding, a specific instance of a projected Landweber (PL) algorithm
- PL algorithms are
  - Fast
  - Easy to implement
  - Flexible—can add other criteria
- Most common PL approach: iterated hard thresholding (IHT)

Project Landweber (PL) Recovery

Iterated Hard Thresholding (IHT)

Given initial transform-coefficient approximation $\check{\mathbf{x}}^{(0)}$:

$$\check{\check{\mathbf{x}}}^{(i)} = \check{\mathbf{x}}^{(i)} + \frac{1}{\gamma}\mathbf{\Psi}\mathbf{\Phi}^T\left(\mathbf{y} - \mathbf{\Phi}\mathbf{\Psi}^{-1}\check{\mathbf{x}}^{(i)}\right),$$

$$\check{\mathbf{x}}^{(i+1)} = \begin{cases} \check{\check{\mathbf{x}}}^{(i)}, & \left|\check{\check{\mathbf{x}}}^{(i)}\right| \geq \tau^{(i)}, \\ 0 & \textbf{else} \end{cases}$$

where

γ : **scaling factor**

$\tau^{(i)}$: **threshold for iteration i**

CS Acquisition of 2D Images

CS Acquisition of 2D Images

- Significant interest in CS for 2D imagery
- CS promises digital cameras:
  - Smaller
  - Cheaper
  - Broader spectral range

Single-Pixel Camera

Takhar *et al.*, *SPIE EI 2006*

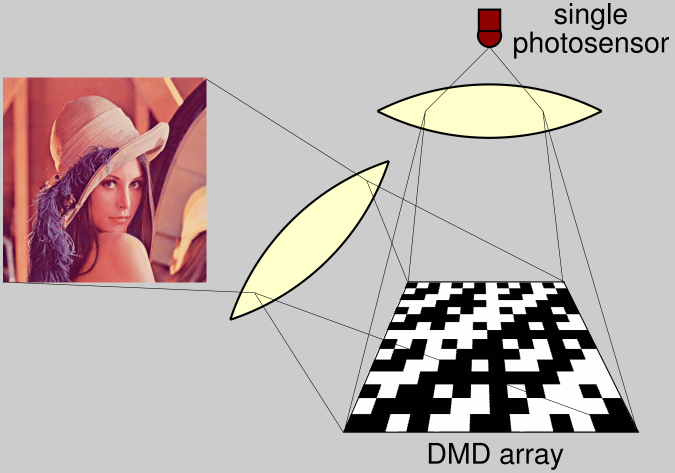
- Uses digital micromirror device (DMD) to optically perform inner products in measurement process
- DMD can effectuate a ± 1 Rademacher measurement matrix

CS Acquisition of 2D Images

BCS of Images and Video

J. E. Fowler

Single-Pixel Camera



Block-Based CS (BCS) for 2D Images

Straightforward CS for 2D Images

Straightforward application of CS to 2D images:

- “Rasterize” $N \times N$ image $\mathbf{X}$ into N^2-dimensional vector $\mathbf{x}$
- Apply $M \times N^2$ measurement matrix $\mathbf{\Phi}$
- Apply 1D CS recovery algorithm (BP, GPSR, PL, etc.) with $\mathbf{\Phi}$
  - Use $N^2 \times N^2$ transform $\mathbf{\Psi}$ (1D representation of a 2D transform)
- “Unrasterize” $\hat{\mathbf{x}}$ to produce image $\hat{\mathbf{X}}$

Block-Based CS (BCS) for 2D Images

Straightforward CS for 2D Images

Problems:

- Computationally expensive reconstruction
- Huge memory to store random sampling operator, $O(N^4)$
- Recovery is "blind" to the fact that data is an image:
  - Searches simply for consistent, sparse solution
  - Not necessarily visually pleasing
  - Ignores known attributes of images, like smoothness

Block-Based CS (BCS) for 2D Images

Solution: Block-Based Compressed Sensing (BCS)

- Image partitioned into $B \times B$ blocks
- $\mathbf{x}_j$: block j of image
- Measurements:

$$\mathbf{y}_j = \mathbf{\Phi}_B \mathbf{x}_j$$

$\mathbf{\Phi}_B$: $M_B \times B^2$ random matrix

- The global measurement matrix is then block-diagonal:

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{\Phi}_B & 0 & \cdots & 0 \\ 0 & \mathbf{\Phi}_B & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & \mathbf{\Phi}_B \end{bmatrix}$$

Block-Based CS (BCS) for 2D Images

Recovery of BCS-Acquired Images

Possible approaches:

- Recover blocks independently—bad idea; severe blocking
- Apply BP-based ℓ_1 recovery (or fast variant) with block-diagonal Φ—does not exploit image properties

Better Approach: BCS-TV

Candès, Romberg, & Tao, *CPAM 2006*

- BP-based ℓ_1 recovery using total variation (TV)
- Implicitly imposes smoothness by pursuing sparsity in the domain of a discrete gradient

BCS-SPL for 2D Images

Our Preferred Approach: BCS-SPL

Gan, *DSP 2007*

- Couple PL reconstruction with a smoothing operator
- Very fast
- Practical—scales well with image size
- Good visual quality
- Block CS with smoothed PL (BCS-SPL)

SPL Reconstruction

- Adds Wiener filter to remove blocking artifacts
- Algorithm:

$$
\begin{aligned}
&\textit{function}\ \mathbf{x}^{(i+1)} = \mathrm{SPL}(\mathbf{x}^{(i)}, \mathbf{y}, \mathbf{\Phi}_B, \mathbf{\Psi}, \lambda)\\
&\quad \hat{\mathbf{x}}^{(i)} = \mathrm{Wiener}(\mathbf{x}^{(i)})\\
&\quad \textit{for each block } j\\
&\qquad \hat{\hat{\mathbf{x}}}_j^{(i)} = \hat{\mathbf{x}}_j^{(i)} + \mathbf{\Phi}_B^T(\mathbf{y} - \mathbf{\Phi}_B\hat{\mathbf{x}}_j^{(i)})\\
&\quad \check{\check{\mathbf{x}}}^{(i)} = \mathbf{\Psi}\hat{\hat{\mathbf{x}}}^{(i)}\\
&\quad \check{\mathbf{x}}^{(i)} = \mathrm{Threshold}(\check{\check{\mathbf{x}}}^{(i)}, \lambda)\\
&\quad \bar{\mathbf{x}}^{(i)} = \mathbf{\Psi}^{-1}\check{\mathbf{x}}^{(i)}\\
&\quad \textit{for each block } j\\
&\qquad \mathbf{x}_j^{(i+1)} = \bar{\mathbf{x}}_j^{(i)} + \mathbf{\Phi}_B^T(\mathbf{y} - \mathbf{\Phi}_B\bar{\mathbf{x}}_j^{(i)})
\end{aligned}
$$

- Linear initialization: $\mathbf{x}_j^{(0)} = \mathbf{\Phi}_B^T\mathbf{y}_j$

BCS-SPL for 2D Images

SPL Reconstruction

Attempts to impose:

- Consistency with observations (Landweber step)
- Sparsity (thresholding)
- Smoothness (Wiener filtering)

Advantages

- Simple implementation
- Easy to extend:
  - Redundant, directional transforms
  - More sophisticated thresholding/shrinkage
  - See poster tomorrow. . .

Results for 2D Images

Comparisons—512×512 images; 32×32 blocks

- BCS-SPL with popular transforms:
  - BCS-SPL-DWT
  - BCS-SPL-DCT
- BP-based ℓ_1 reconstruction with total-variation smoothing:
  - BCS-TV
- Fast BP-based reconstruction (uses 2D DWT as sparsity basis Ψ; no smoothing):
  - BCS-GPSR-DWT

Results for 2D Images

Lenna

Algorithm	*Subrate* (M/N)				
	10%	20%	30%	40%	50%
BCS-SPL-DWT	27.8	30.9	32.9	34.6	36.2
BCS-SPL-DCT	27.2	30.2	32.2	34.1	35.7
BCS-TV	27.9	30.6	32.6	34.3	35.9
BCS-GPSR-DWT	22.7	26.0	28.1	29.9	31.3

Results for 2D Images

Lenna for subrate $= 20\%$

BCS-SPL-DWT
30.9 dB

BCS-SPL-DCT
30.3 dB

BCS-TV
30.6 dB

BCS-GPSR-DWT
26.0 dB

Results for 2D Images

Observations

Reconstruction quality:

- BCS-SPL-DWT, BCS-SPL-DCT, and BCS-TV close in performance
- BCS-GPSR-DWT significantly worse

Execution times:

- BCS-SPL-DWT, BCS-SPL-DCT: 2–3 min.
- BCS-GPSR-DWT: 20–50 sec.
- BCS-TV: 3–4 hrs.

CS for Video

CS Acquisition for Video

- BCS samples every frame identically, e.g., using single-pixel camera frame-by-frame
- A 3D sampling of video "volume" impractical

Straightforward Reconstruction

- Reconstruct 2D frames independently using 2D transform Ψ
- Reconstruct 3D "volume" using 3D transform Ψ (e.g., Wakin *et al.*, *PCS 2006*)
- Neither exploits temporal redundancies due to frame-to-frame motion
- Memory and computational complexity of 3D recovery is substantial

CS for Video

Motion-Compensated CS Reconstruction

- Use neighboring frame(s) to make motion-compensated prediction, $\mathbf{x}_c$, of current frame $\mathbf{x}$
- Modify CS reconstruction of $\mathbf{x}$ to use $\mathbf{x}_c$

Approach 1: Initialization

Kang & Lu, *ICASSP 2009*

- 2D CS reconstruction initialized using $\mathbf{x}_c$ rather than usual initialization (e.g., $\Phi^T\mathbf{y}$)
- Works for any single-frame CS reconstruction
- We use BCS-SPL:
  - MC-BCS-SPL (Initialization)

CS for Video

Approach 2: Residual Reconstruction

Mun & Fowler, to be submitted 2010

- Apply CS reconstruction to motion-compensated **residual**
- Residual should be much sparser than original frame
- Works for any single-frame CS reconstruction
- We use BCS-SPL:
  - MC-BCS-SPL (Residual Reconstruction)

MC-BCS-SPL (Residual Reconstruction)

Given:

- Reference frame, $\mathbf{x}_r$
- Motion-vector field, MV
- Block-based measurements of current frame, $\mathbf{y} = \mathbf{\Phi}\mathbf{x}$
- 2D transform $\mathbf{\Psi}$

Algorithm:

- Motion-compensated frame: $\mathbf{x}_c = \mathrm{MC}(\mathbf{x}_r, MV)$
- Projected residual: $\mathbf{r} = \mathbf{y} - \mathbf{\Phi}\mathbf{x}_c$
- Reconstructed residual: $\hat{\mathbf{r}} = \mathrm{SPL}(\mathbf{r}, \mathbf{\Phi}, \mathbf{\Psi})$
- Reconstructed current frame: $\hat{\mathbf{x}} = \mathbf{x}_c + \hat{\mathbf{r}}$

Single-Frame Results for MC-BCS-SPL

Football—SIF frame, quarter-pixel ME, subrate $= 10\%$

BCS-SPL-DCT	MC-BCS-SPL-DCT	MC-BCS-SPL-DCT
23.9 dB	Initialization	Residual
	24.4 dB	26.2 dB

Single-Frame Results for MC-BCS-SPL

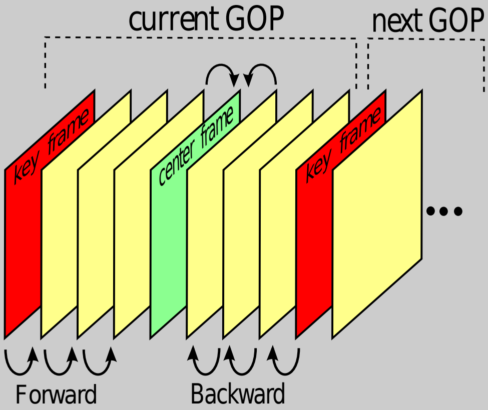
Susie—SIF frame, quarter-pixel ME, subrate $= 10\%$

BCS-SPL-DCT 28.4 dB	MC-BCS-SPL-DCT Initialization 30.3 dB	MC-BCS-SPL-DCT Residual 40.5 dB

Multiple-Frame MC-BCS-SPL

GOP: 8 frames

- **Keyframe: sampled with high subrate**
- **Non-keyframes: sampled with low subrate**



Multiple-Frame MC-BCS-SPL

Idea: “Bootstrap” determination of motion fields

- Reconstruct each frame of GOP individually with BCS-SPL
- Determine motion fields for reconstructed frames
- For each non-keyframe $\hat{\mathbf{x}}$:
  - Use MC-BCS-SPL to redo reconstruction $\hat{\mathbf{x}}$
  - Estimate new motion field using new $\hat{\mathbf{x}}$
  - Repeat. . .
- First half of GOP predicted in forward direction
- Second half of GOP predicted in backward direction (start with keyframe of next GOP)
- Iterative reconstruction of center frame alternates between directions

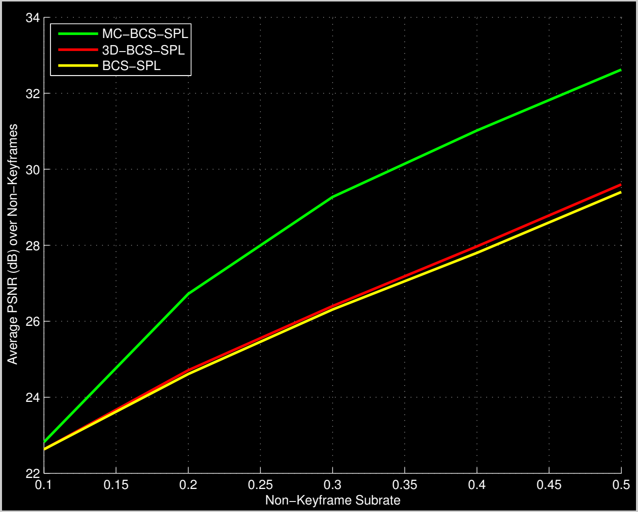
Results for Video

Comparisons

- BCS-SPL
  - Independent frame-by-frame BCS-SPL
- MC-BCS-SPL
  - Residual reconstruction
  - Forward/backward GOP processing
- 3D-BCS-SPL
  - Video "volume" partitioned into 3D blocks
  - BCS-SPL reconstruction uses block-based 3D transform
  - No motion compensation
- All techniques use:
  - Same frame-by-frame 2D block-based sampling
  - Block DCT transform (2D or 3D)
  - Keyframe subrate = 70%

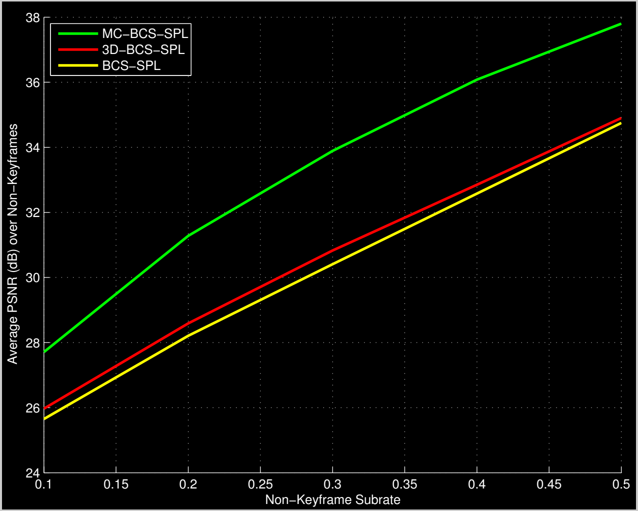
Results for Video

Coastguard: 296 frames



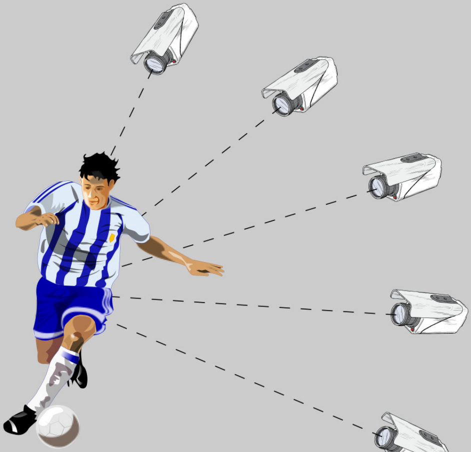
Results for Video

Foreman: 88 frames



Multiview Image Sets

Multiview Image Acquisition



Multiview Image Sets

Middlebury Multiview Database

Monopoly

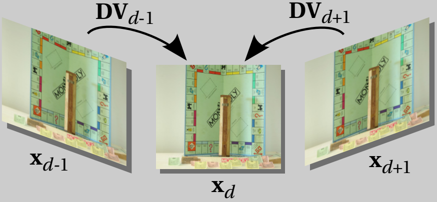
Aloe

Disparity-Compensated BCS-SPL (DC-BCS-SPL)

DC-BCS-SPL

Trocan *et al.*, *ICME 2010 & ICIP 2010*

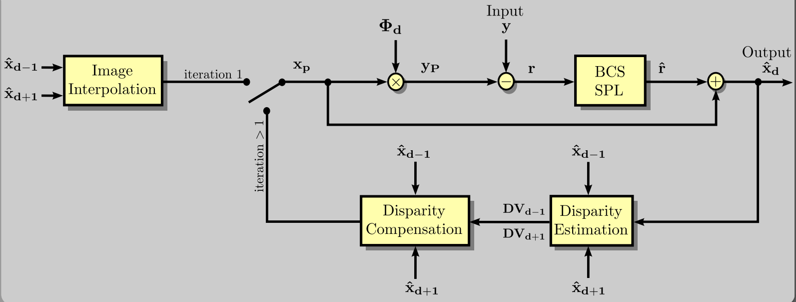
- Adapt MC-BCS-SPL to multiview scenario
- Predict current image $\mathbf{x}_d$ using disparity compensation (DC) between:
  - reconstructed left image, $\hat{\mathbf{x}}_{d-1}$
  - reconstructed right image, $\hat{\mathbf{x}}_{d+1}$
  - disparity-vector (DV) fields, $\mathbf{DV}_{d-1}$ and $\mathbf{DV}_{d+1}$



- BCS-SPL reconstruction from DC residual

DC-BCS-SPL

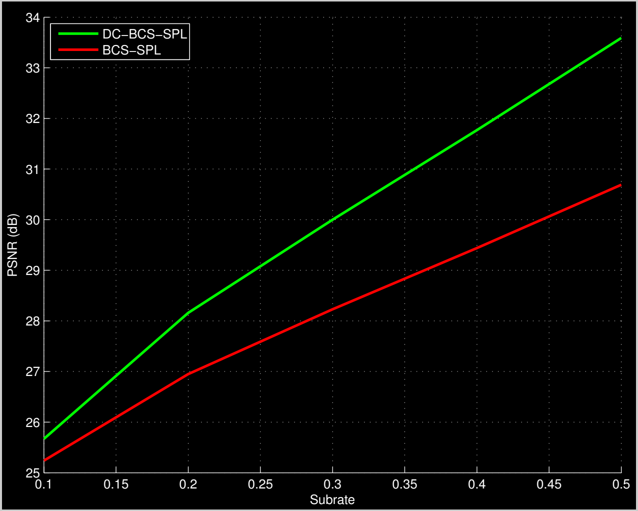
Algorithm



- All images of multiview set reconstructed individually with BCS-SPL
- DV determined from reconstructed images

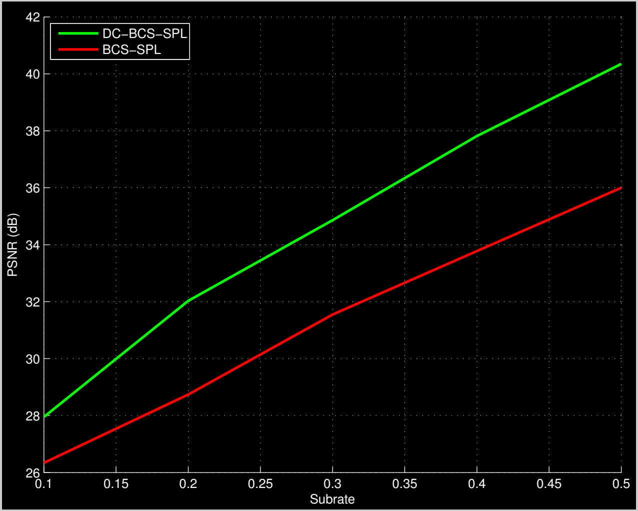
Results for Multiview

Aloe



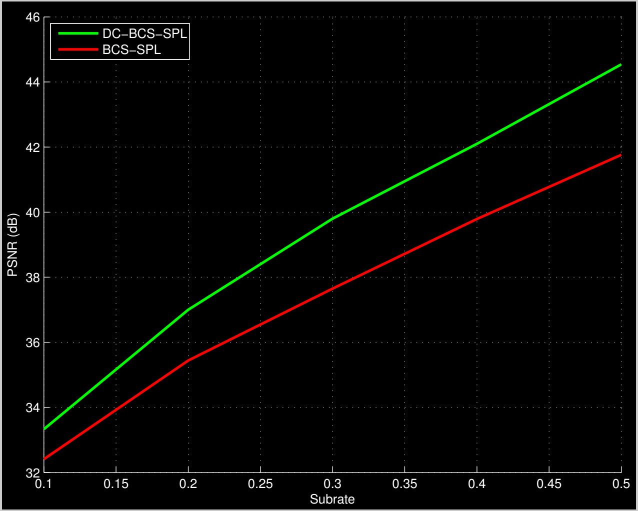
Results for Multiview

Monopoly



Results for Multiview

Bowling



Perspectives

General Observations

- Sparsity alone is not sufficient for image reconstruction with good visual quality
- Reconstruction should capitalize on known properties/processes for imagery (smoothness, motion compensation, . . .)
- It is easy to incorporate image-relevant criteria into the Projected Landweber (PL) formulation

Perspectives

Caveat—CS is not Compression

- It is tempting to couple random CS projections with scalar quantization to produce a “compressed” bitstream
- “[. . .] compressive sampling combined with ordinary quantization is a bad compression technique”—Goyal, Fletcher, Rangan, *SP Magazine 2008*
- CS really makes sense only as dimensionality reduction that takes place simultaneously with data acquisition within the sensing device
  - reduce sensing cost when each sample is expensive to acquire
  - reduce storage/transmission cost in severely resource-constrained sensors

For Further Information...

References

- S. Mun and J. E. Fowler, "Block Compressed Sensing of Images Using Directional Transforms," *ICIP 2009*
- M. Trocan, T. Maugey, J. E. Fowler, and B. Pesquet-Popescu, "Disparity-Compensated Compressed-Sensing Reconstruction for Multiview Images," submitted to *ICME 2010*
- M. Trocan, T. Maugey, E. W. Tramel, J. E. Fowler, B. Pesquet-Popescu, "Compressed Sensing of Multiview Images Using Disparity Compensation," submitted to *ICIP 2010*
- S. Mun and J. E. Fowler, "Residual Reconstruction for Block-Based Compressed Sensing of Video," to be submitted

For Further Information. . .

MATLAB Source Code

- **BCS-SPL Version 1.2**

  `http://www.ece.msstate.edu/~fowler/`