Recursive Low-rank and Sparse Recovery of Surveillance Video using Compressed Sensing

Shuangjiang Li, Hairong Qi

Department of Electrical Engineering and Computer Science
University of Tennessee, Knoxville

Oct. 1, 2014

8th ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC), Venezia, Italy

Outline

Outline

Background on SCNs

Smart Camera Networks (SCNs) have been traditionally used in surveillance and security applications, where a plural of cameras are deployed and networked with each other through wireless connections.

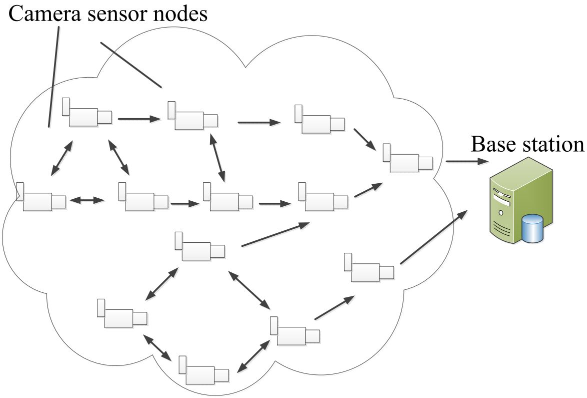


Figure: An illustration of SCNs.

Background on SCNs (cont'd)

- The ability to detect anomalies and moving objects in a scene automatically and quickly is of particular interest.
- Detection of moving objects is a well-established problem. (e.g., background subtraction, object segmentation, and sequential estimation for the objects of interest)
- Due to the growing availability of cheap, high-quality cameras, the amount of data generated by the video surveillance system has grown drastically.
- The challenge arises on how to process, store or transmit such enormous amount of data under real-time and bandwidth constraints.

Motivation

-A Compressed Sensing(CS) approach for SCNs object detection

- Multiple number of cameras with wireless connection need transmit surveillance videos to a processing center, at the same time, most of the data is uninteresting due to inactivity (e.g., background).
- It is imperative for SCNs to transmit a small amount of data with enough information for reliable detection and tracking of moving objects or anomalies.
- CS approach for object detection.

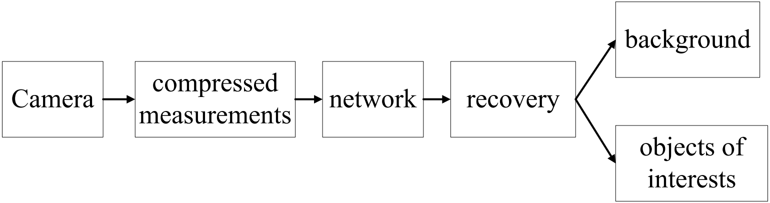


Figure: Compressed sensing recovery for object detection in SCNs framework.

Motivation

-A Compressed Sensing(CS) approach for SCNs object detection

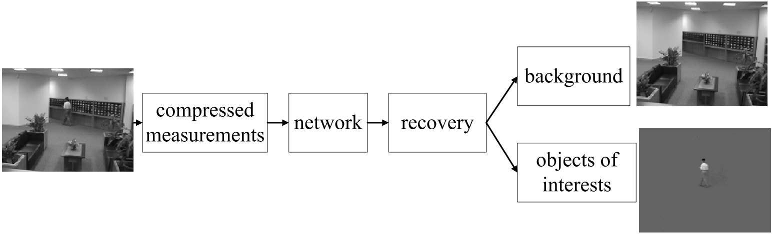


Figure: Compressed sensing recovery for object detection in SCNs framework.

- The reconstructed video consists of a low-rank part which corresponds to the background and the sparse part, which is the object of interest.
- CS recovery on a single frame for initial estimation, then recursively recover the low-rank and sparse component in the entire video.

Background on CS Recovery

-Signal Sparse Coding/Representation and Recovery

Assume a signal $x \in \mathcal{R}^N$ can be represented as $x = \Psi\alpha$, where $\Psi \in \mathcal{R}^{N\times M}(N < M)$ is a basis or an over-complete dictionary, and most entries of the coding vector α are zero or close to zero.

$$\boxed{\alpha_x = \arg\min_{\alpha}\{\|x - \Psi\alpha\|_2^2 + \lambda_\alpha\|\alpha\|_1\}}$$

In CS recovery, what we observe is the projected measurement y via $y = \Phi x + \nu$. Needing to solve,

$$\boxed{\hat{\alpha} = \arg\min_{\alpha}\{\|y - \Phi\Psi\alpha\|_2^2 + \lambda_\alpha\|\alpha\|_1\}}$$

then x is reconstructed by $\hat{x} = \Psi\hat{\alpha}$.

Problem Formulation

-Foreground and Background of a Frame

A video sequence consists of a number of frames (i.e., images).

- Let $x_t \in \mathbb{R}^{m \times n}$ be a vector formed from pixels of frame t of the video sequence, for $t = 1, \cdots, T$, where T is the total number of frames, m and n are the dimensions of each frame.
- The current frame x_t, is an overlay of foreground image, F_t, over the background image, B_t.
- The goal is to recover both F_t and B_t at each time frame t in real-time.
- Many foreground pixels are zero and hence F_t is a sparse matrix. We let T_t denote the foreground support set, i.e., $T_t := \{i : (F_t)_i \neq 0\}$.

$$(x_t)_i := \begin{cases} (F_t)_i & \text{if } i \in T_t \\ (B_t)_i & \text{otherwise} \end{cases}$$

Problem Formulation

-CS Recovery on a Single Frame

- Assume, each frame can be re-arranged as an $N \times 1$ vector (i.e., $N = m \times n$). Let Φ_t be an $M \times N$ CS measurement matrix, where $M < N$.

$$y_t = \Phi_t x_t \tag{1}$$

  where y_t is a vector of length M.

- To recover x_t from y_t, first y_t is sparsely coded with respect to the basis $\Psi \in \mathbb{R}^{N \times N}$ by solving the following minimization problem

$$\hat{\alpha} = \arg\min_{\alpha} \{ \| y_t - \Phi_t \Psi \alpha \|_2^2 + \lambda_\alpha \| \alpha \|_1 \} \tag{2}$$

  and then x_t is reconstructed by $\hat{x}_t = \Psi \hat{\alpha}$.

Problem Formulation

-Low-rank and Sparse Components of a Frame

- Let μ_t denote the mean background image and let $L_t := B_t - \mu_t$

$$(S_t)_i := \begin{cases} (F_t - B_t)_i = (F_t - \mu_t - L_t)_i & \text{if } i \in T_t \\ 0 \quad \text{otherwise} \end{cases} \tag{3}$$

- Let $M_t := \hat{x}_t$ be the frame t reconstructed from CS recovery algorithm with mean subtracted, then

$$M_t := S_t + L_t \tag{4}$$

- Here, S_t is a sparse vector with support set T_t, and L_t are dense matrices lie in a slowly changing low dimensional subspace.

Outline

Section Outline

We propose the recursive Low-rank and Sparse Recovery using Douglas-Rachford splitting (rLSDR) that consists of three major components.

- Component 1: Single frame recovery
  - CS image recovery
  - Nonlocal means filtering
  - Nonlocal Douglas-Rachford splitting (NLDR) algorithm
- Component 2: Fast low-rank background initialization
- Component 3: Recursive sparse recovery and background update

CS Image Recovery

Direct approach: reshape 2D images in 1D vector

- the curse of dimensionality (e.g., a 512×512 image $\Rightarrow 262,144$ dim.). Computational Complexity!!
- need to store a large random measurement operator (e.g., $\Phi \in \mathcal{R}^{0.3*262,144 \times 262,144}$). Storage Problems!!

"Divide and conquer" approach

The image is divided into small patches with size of $B \times B$, and sampled with *the same random measurement operator*[a].

- lose the global structure of an image
- cause blocking artifacts and need extra smoothing process
- result in low recovery PSNR

[a] Lu Gan,"Block compressed sensing of natural images," in International Conference on Digital Signal Processing, IEEE, 2007

The Proposed NLDR Algorithm

We propose the NLDR (NonLocal Douglas-Rachford Splitting) for SCNs CS image recovery.

- Block-based approach (using Iterative Soft Thresholding[1]) to reconstruct the image first (intermediate result).
- Instead of treating each block as a separate/individual sub-CS recovery task. We propose to group similarity patches into a low-rank patch matrix and conduct low-rank estimation (i.e., denoising to prevent the noise from accumulating).
- Each denoised patch is then combined with CS measurement constraints to further improve the frame recovery result.
- We propose to solve the above problem using Douglas-Rachford Splitting method.

[1] I. Daubechies et al., An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, 2004

Nonlocal Means Filtering

- Take advantage of the high degree of redundancy/self-similarities of any natural image for denoising purpose by Buades [2].
- Given two image patches centered at pixel p_i and p_j, we calculate the similarity of the intensity gray level within a window size $B \times B$.

$$\omega_{ij} = \frac{1}{c_i} \exp\left(\frac{-\|p_i - p_j\|_2^2}{h^2}\right), \ j = 1, \cdots, q \tag{5}$$

q is the number of similar patches, h is scalar and c_i is the normalization factor.

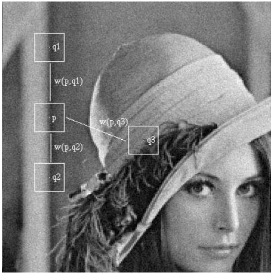


Figure: The illustration of the nonlocal means filtering.

[2] Buades et al. "A review of image denoising algorithms, with a new one," Multiscale Mod. & Simu., 2005

Patch Denoising using Low-rank Approximation

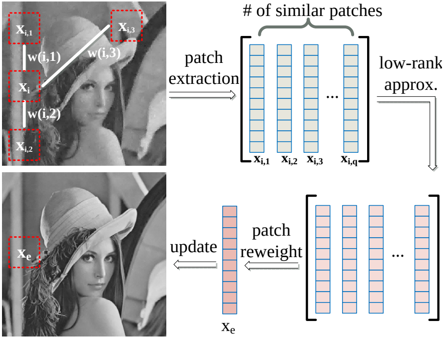


Figure: An illustration of nonlocal estimation and similar patches denoising using low-rank matrix approximation.

$$\boxed{\min_{\hat{B}_i} \frac{1}{2}\|\hat{B}_i - B_i\|_2^2 + \lambda_{B_i}\|\hat{B}_i\|_*}$$

- $\|\hat{B}_i\|_*$ is the nuclear norm
- $\|\hat{B}_i\|_* \triangleq \text{trace}(\sqrt{\hat{B}_i{}^T \hat{B}_i}) = \sum_{r=1}^{q} \sigma_r$, where σ_r are the singular values of $\hat{B}_i$.

From Patch to Image

- For each patch p_i, we have a set of its similar patches, denoted by Ω_i. Then the nonlocal estimates of each patch $\hat{p}_i$ can be computed as $\hat{p}_i = \sum_{j \in \Omega_i} \omega_{ij} p_{i,j}$
- Further, this can be written in a matrix form as

$$\hat{x}_i \doteq \mathbf{W} \sum \hat{p}_i, \ \mathbf{W}(i,j) = \begin{cases} \omega_{ij}, & \text{if } x_j \in \Omega_i \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

where $\hat{x}_i$ is the nonlocal estimated single video frame output.

Incorporating CS Measurement Constraint

-From Patches to Image

- Since the columns of B_i (or patches) are also a subset of the reconstructed image from IST recovery algorithm, it should subject to the CS measurement constraint $y = \Phi x$.
- Multiple $\mathbf{W}$ on both sides

$$\boxed{\min_{\hat{B}_i} \frac{1}{2}\|\hat{B}_i - B_i\|_2^2 + \lambda_{B_i}\|\hat{B}_i\|_*}$$

- We formulate the denoising problem as

$$\boxed{\min_{x} \frac{1}{2}\|x - \mathbf{W}B_i\|_2^2 + \lambda_x\|x\|_* \text{ s.t. } y = \Phi x} \tag{7}$$

- $\mathbf{W}$ is a patch reweighting matrix

Douglas-Rachford Splitting

We use Douglas-Rachford Splitting (DR) to solve Eq (7).

- Commonly use gradient or projection based method (i.e., POCS).
- DR uses *proximity operator* prox_f (extend the projection operator to general case), for example, soft thresholding operations can not be solved using convex projection.
- DR is an iterative scheme to minimize two convex functions with rate of convergence $\mathcal{O}(1/k), k$ is iteration number.
- We propose to use DR and divided the problem into two convex functions (where the *proximity operators* are available) and solved the problem.

Douglas-Rachford Splitting

- $F(x) = \iota_{\mathcal{C}}(x)$ and $G(x) = \|x\|_*$, where $\mathcal{C} = \{x : \Phi x = y\}$ and $\iota_{\mathcal{C}}$ is the indicator function

$$\boxed{\min_x \frac{1}{2}\|x - \mathbf{W}B_i\|_2^2 + \lambda_x\|x\|_* \text{ s.t. } y = \Phi x}$$

Douglas-Rachford Splitting

-Proximity Operators

- Given that $F(x) = \iota_{\mathcal{C}}(x)$, $\text{prox}_{\gamma\iota_{\mathcal{C}}F}(x)$ is the same as projections onto convex sets (POCS), and does not depends on γ.

$$\boxed{\text{prox}_{\gamma\iota_{\mathcal{C}}F}(x) = \text{prox}_{\iota_{\mathcal{C}}F}(x) = x + \Phi^{+}(y - \Phi x)}$$

  where $\Phi^{+} = \Phi^{T}(\Phi\Phi^{T})^{-1}$.

- The proximal operator of $G(x)$ is the soft thresholding of the singular values

$$\boxed{\text{prox}_{\gamma G}(x) = U(x)\rho_{\lambda_x}(S(x))V(x)^{*}}$$

  where $x = USV^{*}$ is the singular value decomposition of the matrix x and $S = \text{diag}(s_i)_i$ is the diagonal matrix of singular values s_i.
  $\rho_{\lambda_x}(S(x))$ is defined as a diagonal operator.

$$\boxed{\rho_{\lambda}(S) = \text{diag}(\max(0, 1 - \lambda_x/|s_i|)s_i)_i}$$

Douglas-Rachford Splitting

-Iterations

- We can then solve the problem in Eq (7) using the Douglas-Rachford iterations given by

$$\boxed{\tilde{x}_{k+1} = (1 - \frac{\mu}{2})\tilde{x}_k + \frac{\mu}{2}\text{rprox}_{\gamma G}(\text{rprox}_{\gamma F}(\tilde{x}_k))} \tag{8}$$

- Then the $(k+1)$-th solution $\hat{x}_{k+1}$ is calculated by $\hat{x}_{k+1} = \text{prox}_{\gamma F}(\tilde{x}_{k+1})$, where $\tilde{x}_0$ is the output of IST recovery result.
- Here the reversed-proximal mappings is given by $\text{rprox}_{\gamma F} = 2\text{prox}_{\gamma F} - x$ for $F(x)$ and $G(x)$ respectively.
- Set $\lambda_x > 0$ and $0 < \mu < 2$ which guarantee $\hat{x}$ is a minimizer.

Single Frame Recovery Algorithm - NLDR

Algorithm 1: NLDR Algorithm

Input:

- CS Measurement matrix: $\Phi_t \in \mathbb{R}^{M \times N}$
- Basis matrix: $\Psi \in \mathbb{R}^{N \times N}$
- Measurements: $y_t \in \mathbb{R}^{M}$
- \# of iterations: iter.

Output:

- An estimate $\hat{x}_t \in \mathbb{R}^{N}$ of the original single frame x_t.

```
Obtain an initial recovery x̂_IST from IST
Initialize x̂_nl ← x̂_IST
Calculate nonlocal weights ω_ij using Eq. (5)
Update x̂_nl ← W x̂_i using Eq. (6)
for k = 0, 1, 2, ··· , iter do
    Initialize x̃_0 ← x̂_nl
    Calculate x̃_{k+1} using Eq. (8)
end for
return x̂_t ← x̃_{k+1}
```

Obtain an initial recovery $\hat{x}_{IST}$ from **IST**
Initialize $\hat{x}_{nl} \leftarrow \hat{x}_{IST}$
Calculate nonlocal weights ω_{ij} using Eq. (5)
Update $\hat{x}_{nl} \leftarrow \mathbf{W}\hat{x}_i$ using Eq. (6)
for $k = 0, 1, 2, \cdots$, iter **do**
    Initialize $\tilde{x}_0 \leftarrow \hat{x}_{nl}$
    Calculate $\tilde{x}_{k+1}$ using Eq. (8)
end for
return $\hat{x}_t \leftarrow \tilde{x}_{k+1}$

Low-rank Component Initialization

- The second component of the proposed rLSDR algorithm is to estimate the low-rank background image based on a few recovered video frames. (e.g., first 50 frames)
- The common approach would be applying SVD on the recovered video frames
  - Performing SVD operation is usually very time-consuming, especially for large resolution video frames which hinders the "on-the-fly" estimation.
  - Often we just need a rough estimation of the low-rank component which can later be refined upon receiving new video frames.

Bilateral Random Projections (BRP) based Low-rank approximation

- Given r bilateral random projections of a $p \times q$ dense matrix X (w.l.o.g, $p \geq q$), i.e., $U = XA_1$ and $V = X^T A_2$, where $A_1 \in \mathbb{R}^{q \times r}$ and $A_2 \in \mathbb{R}^{p \times r}$ are random matrices,

$$\boxed{L = U(A_2^T U)^{-1} V^T} \tag{9}$$

  is a fast rank-r approximation of X.
- The L in Eq. (9) has been proposed by Fazel et al. [3] as a recovery of a rank-r matrix X from U and V, where A_1 and A_2 are independent Gaussian or subsampled Fourier random matrices.

[3] Fazel et al. Compressed Sensing and Robust Recovery of Low Rank Matrices, 2008

Recursive Sparse Recovery and Low-rank Updates

- After the low-rank background component L_t has been estimated, we then recursively update the sparse component and background estimation upon receiving the CS measurements y_{t+1} of new frame x_{t+1}.
- The CS recovered new frame $\hat{x}_{t+1}$ is obtained using the proposed NLDR algorithm.
- The sparse recovery problem to find S_{t+1} can be formulated as follows

$$\begin{aligned} &\min_{S_{t+1}} \frac{1}{2}\|\hat{x}_{t+1} - L_t - S_{t+1}\|_2^2 + \lambda_s \|S_{t+1}\|_1 \\ &\text{s.t.} \quad \|y_{t+1} - \Phi_{t+1}(L_t + S_{t+1})\|_2^2 \leq \epsilon \end{aligned} \tag{10}$$

  where L_t is estimated background at the frame t.
- The only unknown in Eq. (10) is S_{t+1}, which can be solved using NLDR algorithm to estimate $\hat{S}_{t+1}$.

Summary of the rLSDR Algorithm

Algorithm 2: rLSDR Algorithm

Input:

- CS Measurement matrix: $\Phi_t \in \mathbb{R}^{M \times N}$,
- Measurements data matrix: $y_t \in \mathbb{R}^{M \times p}$
- Initialize random matrices: A_1, A_2,
- Number of training frames: trn.

Output:

- CS recovered frames: $\hat{x} \in \mathbb{R}^{N \times p}$,
- Background and object estimate: $\hat{L}, \hat{S}$.

Step 1: Initial frame recovery
for $i = 1, \cdots,$ trn do
  $X(1 :$ trn$) \leftarrow$ **NLDR**(y_i)
end for
Step 2: Background initialization
Estimate L using Eq. (9)
Step 3: Recursive update L and S
for $t =$ trn$, \cdots ,$ p do
  Frame recovery: $\hat{x}_{t+1} \leftarrow$ **NLDR**(y_{t+1})
  Sparse est.: Solve Eq. (10) for $\hat{S}_{t+1}$ using **NLDR**
  Calculate L_{t+1}: $L_{t+1} = \hat{x}_{t+1} - \hat{S}_{t+1}$, update Eq. (9)
  Background est.: $\hat{L}_{t+1} = L(t+1)$
end for
return $\hat{x}, \hat{L}, \hat{S}$

Outline

Experimental Results

-Experiments Settings

- We apply **rLSDR** to two surveillance videos [4] *Restaurant* and *Curtain*.
- *Curtain* consists of 304 frames each of dimension 64×80, *Restaurant* contains 200 frames with dimension 144×176.
- We first experiment on the single frame recovery result by comparing **NLDR** with two popular CS image recovery algorithms, BCS-SPL [5] and TVNLR [6].
- We then experiment on video object detection and compare results with Principal Component Pursuit (**PCP**) [7] and **ReProCS** (using ADMM) [8].

---

[4] http://perception.i2r.a-star.edu.sg/bk_model/bk_index.html

[5] J. Fowler, Block compressed sensing of images using directional transforms. 2009.

[6] J. Zhang, Improved total variation based image compressive sensing recovery by nonlocal regularization. 2013.

[7] Candès et al. Robust principal component analysis? 2011

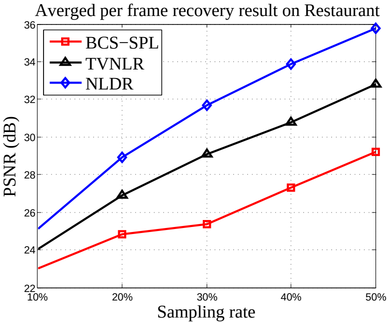
[8] C. Qiu, ReProCS: A missing link between recursive robust PCA and recursive sparse recovery in large but correlated noise. 2011

Experimental Results

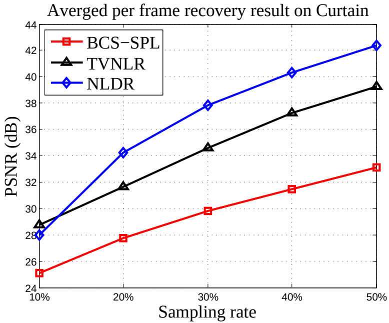
-Experiments Settings (cont'd)

- The block-based image patch is of size 6×6. We set the number of similar patches q in the nonlocal estimation step as 45.
- We use the scrambled Fourier matrix as the CS measurement matrix Φ and DCT matrix as the basis Ψ to represent the original image in the initial IST recovery.
- The parameter is selected as $\mu = 1$ for DR iteration and $\lambda_f = \frac{c_i}{\max(s_i)}$ for each iteration where $c_i = C_0 * \epsilon, 0 < \epsilon < 1$ and C_0 is a constant.

Experimental Results
-PSNR Performance



(a)



(b)

Figure: Averaged per frame recover result comparison on (a) **Restaurant** (b) **Curtain**.

Experimental Results: Background and Object Detection

Figure: First column: original *Restaurant* video frames at t = 70, 116, 140. Second column: frame recovered by **NLDR** with 30% measurements. Next 2 columns: background and object estimated by **rLSDR**. *Restaurant* uses first 50 training frames to initialize the background.

Experimental Results: Object Detection Comparison

(a) Ori. (b) Rec. (c) **rLSDR** (d) **rLSDR** (e) **PCP** (f) **PCP** (g) **ReProCS** (h) **ReProCS**

Figure: First column: original *Curtain* video frames at t = 65, 103, 140. Second column: frame recovered by **NLDR** with 30% measurements. Next 6 columns: background and object estimated by **rLSDR**, **PCP** and **ReProCS** respectively.

Outline

Conclusion

- We presented **rLSDR**, a CS-based surveillance video processing algorithm to recursively estimate the low-rank background and sparse object. The spatial and temporal low-rank features of the video frame were successfully exploited.
- Capitalized on the self-similarities within each spatial frame, we proposed **NLDR** for the single frame CS recovery that had high recovery PSNR under various sampling rates compared with the-state-of-art recovery algorithm.
- We then proposed rLSDR that recursively estimates the background through efficient bilateral random projection (BPR) for background initialization.

Thank you!

Any questions?