Video Background Tracking and Foreground Extraction via L1-subspace Updates

Michele Pierantozzi*, Ying Liu+, Dimitris A. Pados+, Stefania Colonnese*

*Department of Information Engineering, Electronics and Telecommunications,
La Sapienza University of Rome, Rome, Italy 00185

+Department of Electrical Engineering, State University of New York at Buffalo,
Buffalo, NY 14260

ABSTRACT

We consider the problem of online foreground extraction from compressed-sensed (CS) surveillance videos. A technically novel approach is suggested and developed by which the background scene is captured by an L_1-norm subspace sequence directly in the CS domain. In contrast to conventional L_2-norm subspaces, L_1-norm subspaces are seen to offer significant robustness to outliers, disturbances, and rank selection. Subtraction of the L_1-subspace tracked background leads then to effective foreground/moving objects extraction. Experimental studies included in this paper illustrate and support the theoretical developments.

Keywords: Background and foreground extraction, compressive sampling, compressed sensing, convex optimization, feature extraction, L_1 principal component analysis, singular value decomposition, total-variation minimization, video surveillance.

1. INTRODUCTION

An interesting line of research on video background subtraction is based on low-rank subspace approximation. In contrast to pixel-level background modeling schemes[1−],[10] low-rank subspace approximations are block-level or image-level and extensively exploit spatial correlations. Subspace/background estimation is conventionally carried out by L_2-norm based principal component analysis (L_2-PCA), such as block-level one-dimensional PCA[11] and frame-level two-directional two-dimensional PCA ($(2D)^2$PCA).[12] L_2-PCA in general seeks a low-rank subspace to represent the background scene but is easily affected by moving objects in the foreground scene, i.e. "outliers" that are numerically distant from the background. In recent years, there has been a growing interest in robust PCA methods to deal with outliers in principal-component design[13−],[20] such as subspace decomposition via L_1-error minimization[13−],[16] non-negative matrix factorization via Manhattan distance minimization (MahNMF),[17] and robust PCA (RPCA)[18] that performs low-rank background and sparse foreground decomposition. In particular, RPCA minimizes a weighted sum of the nuclear-norm of the low-rank component and the ℓ_1-norm of the sparse component. The robust PCA concept was later adopted in DECOLOR,[19] which in addition models continuity of the sparse components using Markov Random Fields (MRFs) to improve the accuracy of detecting contiguous outliers.

Most existing algorithms[19,20] for subspace-based background extraction are designed for *offline* processing in which video frames are first collected and then processed. In this paper, we use the explicit term background *tracking*, instead of background extraction, because we want to pursue *online* processing in which the decoder in the processing center continuously receives new frames from cameras. Therefore, the objective is to update (track) the background when each new frame arrives at the decoder leading to online foreground moving objects extraction. This can be carried out by computing a sequence of L_1-norm subspaces directly in the CS domain where the low-rank property is preserved if each frame of the video is captured by the same compressed sensing

M.P.: E-mail: michele.pierantozzi@gmail.com
Y.L.: E-mail: yl72@buffalo.edu
D.A.P: E-mail: pados@buffalo.edu
S.C.: E-mail: stefania.colonnese@uniroma1.it

Compressive Sensing V: From Diverse Modalities to Big Data Analytics, edited by Fauzia Ahmad,
Proc. of SPIE Vol. 9857, 985708 · © 2016 SPIE · CCC code: 0277-786X/16/$18 · doi: 10.1117/12.2224956

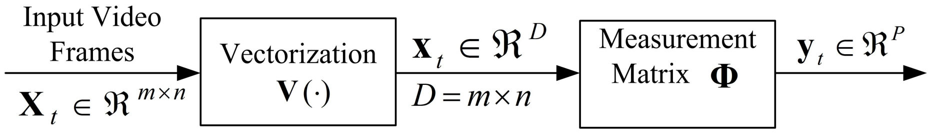

Figure 1. Compressive video encoding block diagram

matrix. The proposed online CS-L_1-subspace update algorithm computes (updates) new L_1 principal components that span the L_1-subspace at each new frame's arrival. Experimental studies that we include in this paper demonstrate the effectiveness of the proposed algorithm and its robustness in the presence of outliers/faulty data in CS measurements.

2. BACKGROUND/FOREGROUND EXTRACTION OF COMPRESSED-SENSED SURVEILLANCE VIDEO

2.1 CS Surveillance Video Acquisition

We consider a practical CS surveillance video acquisition system that performs pure, direct compressed sensing on each video frame. In the simple compressive video encoding block diagram shown in Fig. 1, each frame $\mathbf{X}_t$ of size $m \times n$ is viewed as a vectorized column $\mathbf{x}_t \in \mathbb{R}^D$, $D = mn$, where t is the frame index. Compressive sampling is performed by projecting $\mathbf{x}_t$ onto a $P \times D$ ($P < D$) measurement matrix $\mathbf{\Phi}$,

$$\mathbf{y}_t = \mathbf{\Phi}\mathbf{x}_t, \quad t = 1, 2, ..., \tag{1}$$

where $\mathbf{\Phi}$ is generated by randomly permuting the columns of an order-k, $k \geq D$ and multiple-of-four, Walsh-Hadamard (WH) matrix followed by arbitrary selection of P rows from the k available WH rows (if $k > D$, only D arbitrary columns are utilized). This class of WH measurement matrices has the advantage of easy implementation (antipodal ±1 entries), fast transformation, and satisfactory reconstruction performance as we see later on. A reacher class of matrices can also be found and used[21],[22] For practical implementation, $\mathbf{\Phi}$ is generated once and fixed for all frames in the video sequence. The resulting CS measurement vectors $\mathbf{y}_t$, $t = 1, 2, ...$, are then transmitted to the decoder.

2.2 Background/Foreground Extraction via CS-domain L_1-PCA

Consider a sequence of N video frames in the form of a matrix $\mathbf{X} = [\mathbf{x}_1 \quad \mathbf{x}_2 \quad ... \quad \mathbf{x}_N] \in \mathbb{R}^{D\times N}$. $\mathbf{X}$ can be viewed as the sum of a low rank component $\mathbf{L} \in \mathbb{R}^{D\times N}$ that represents the background scene and a sparse component $\mathbf{E} \in \mathbb{R}^{D\times N}$ that represents moving objects in the foreground scene, $\mathbf{X} = \mathbf{L} + \mathbf{E}$. The CS measurement vectors with sensing matrix $\mathbf{\Phi} \in \mathbb{R}^{P\times D}$ collected at the monitoring center form the matrix

$$\mathbf{Y} = \mathbf{\Phi}\mathbf{X} = \mathbf{\Phi}(\mathbf{L} + \mathbf{E}) = \mathbf{Y}_\mathbf{L} + \mathbf{\Phi}\mathbf{E} \tag{2}$$

where if $\mathbf{L}$ is a rank-d matrix, then $\mathbf{Y}_\mathbf{L} \triangleq \mathbf{\Phi}\mathbf{L}$ is also of rank d and represents the compressed-sensed background scene.

A conventional tool to extract the low-rank component $\mathbf{Y}_\mathbf{L}$ from $\mathbf{Y}$ is L_2-PCA, which seeks the best rank-d ($d \leq \min\{P, N\}$) representation of $\mathbf{Y}_\mathbf{L}$ by solving

$$\mathcal{P}_1^{L_2} : (\mathbf{R}_{L_2}, \mathbf{S}_{L_2}) = \arg \min_{\substack{\mathbf{R}\in\mathbb{R}^{P\times d}, \mathbf{R}^T\mathbf{R}=\mathbf{I}_d \\ \mathbf{S}\in\mathbb{R}^{N\times d}}} \|\mathbf{Y} - \mathbf{R}\mathbf{S}^T\|_2. \tag{3}$$

Problem $\mathcal{P}_1^{L_2}$ is equivalent to the L_2-projection maximization problem

$$\mathcal{P}_2^{L_2} : \mathbf{R}_{L_2} = \arg \max_{\substack{\mathbf{R}\in\mathbb{R}^{P\times d} \\ \mathbf{R}^T\mathbf{R}=\mathbf{I}_d}} \|\mathbf{Y}^T\mathbf{R}\|_2. \tag{4}$$

The optimal $\mathbf{R}_{L_2}$ solution (same in both $\mathcal{P}_1^{L_2}$ and $\mathcal{P}_2^{L_2}$) is simply the d dominant-singular-value left singular vectors of $\mathbf{Y}$.

In practice, however, the observed CS measurements may occasionally be corrupted by outliers $\mathbf{O}$ due to acquisition failures, leading to the acquisition model[23,24]

$$\mathbf{Y} = \mathbf{Y_L} + \mathbf{\Phi E} + \mathbf{O}. \tag{5}$$

In this situation, the problem of background extraction can be tackled by L_1 principal-component analysis (L_1-PCA) that pursues a robust low-rank subspace representation of $\mathbf{Y_L}$ in the form

$$\mathcal{P}^{L_1} : \mathbf{R}_{L_1} = \arg \max_{\substack{\mathbf{R}\in\mathbb{R}^{P\times d} \\ \mathbf{R}^T\mathbf{R}=\mathbf{I}_d}} \|\mathbf{Y}^T\mathbf{R}\|_1. \tag{6}$$

The d columns of $\mathbf{R}_{L_1}$ in (6) are the so-called d L_1 principal components that describe the rank-d subspace in which $\mathbf{Y_L}$ lies. Exact calculation of the L_1 principal components in (6) can be recast as a combinatorial problem[24,].[25] In short, when the rank of the nominal signal $\mathbf{Y_L}$ is $d = 1$, (6) reduces to

$$\mathcal{P}^{L_1} : \mathbf{r}_{L_1} = \arg \max_{\substack{\mathbf{r}\in\mathbb{R}^{P} \\ \|\mathbf{r}\|_2=1}} \|\mathbf{Y}^T\mathbf{r}\|_1 \tag{7}$$

and we can rewrite the maximization problem in (7) as

$$\begin{aligned}
\max_{\substack{\mathbf{r}\in\mathbb{R}^{P} \\ \|\mathbf{r}\|_2=1}} \|\mathbf{Y}^T\mathbf{r}\|_1 &= \max_{\substack{\mathbf{r}\in\mathbb{R}^{P} \\ \|\mathbf{r}\|_2=1}} \max_{\mathbf{b}\in\{\pm1\}^N} \mathbf{b}^T\mathbf{Y}^T\mathbf{r}, && (8)\\
&= \max_{\mathbf{b}\in\{\pm1\}^N} \max_{\substack{\mathbf{r}\in\mathbb{R}^{P} \\ \|\mathbf{r}\|_2=1}} \mathbf{r}^T\mathbf{Y}\mathbf{b}, && (9)\\
&= \max_{\mathbf{b}\in\{\pm1\}^N} \|\mathbf{Y}\mathbf{b}\|_2 \quad \text{and} && (10)\\
\mathbf{b}_{\text{opt}} &= \max_{\mathbf{b}\in\{\pm1\}^N} \mathbf{b}^T\mathbf{Y}^T\mathbf{Y}\mathbf{b}. && (11)
\end{aligned}$$

Therefore, solving (7) is equivalent to finding the optimal binary antipodal vector $\mathbf{b}_{\text{opt}}$ for problem (11), which can be done optimally using exhaustive search in exponential time, or polynomial time,[25] or suboptimally via the single-bit-flipping (SBF) fast algorithm.[26] The SBF algorithm for $d = 1$ starts with an initial binary vector $\mathbf{b}^{(\text{ini})} \in \{\pm1\}^N$ and iteratively produces a sequence of new binary vectors $\mathbf{b}^{(k)},\ k = 1, 2, ...,$ where $\mathbf{b}^{(k+1)}$ at the $(k+1)$-th iteration differs from $\mathbf{b}^{(k)}$ at the k-th iteration only in a single bit position. At each iteration we choose to flip the bit that results in the highest increase of the quadratic value $\mathbf{b}^T\mathbf{Y}^T\mathbf{Y}\mathbf{b}$ in (11). To efficiently identify the bit to flip in a binary vector $\mathbf{b}^{(k)}$ at the k-th iteration we observe that

$$\mathbf{b}^{(k)^T}\mathbf{Y}^T\mathbf{Y}\mathbf{b}^{(k)} = \text{Tr}(\mathbf{Y}^T\mathbf{Y}) + \sum_{i=1}^{N} 2b_i^{(k)} \left\{ \sum_{\substack{j>i \\ j=1}}^{N} b_j^{(k)} (\mathbf{Y}^T\mathbf{Y})_{i,j} \right\}. \tag{12}$$

In view of (12), changing the i-th bit $b_i^{(k)}$ in $\mathbf{b}^{(k)}$ will change the quadratic value by

$$\alpha_i^{(k)} = \pm 4 b_i^{(k)} \left\{ \sum_{\substack{j\neq i \\ j=1}}^{N} b_j^{(k)} (\mathbf{Y}^T\mathbf{Y})_{i,j} \right\}. \tag{13}$$

Therefore, if (13) is negative, changing $b_i^{(k)}$ to $-b_i^{(k)}$ will increase the quadratic value in (12) by $|\alpha_i^{(k)}|$. Then, flipping the bit with the most negative contribution will offer the biggest increase. After we reach a (suboptimal) solution of (11), say $\mathbf{b}^*$, the L_1 principal component $\mathbf{r}_{L_1}$ can be obtained by

$$\mathbf{r}_{L_1} = \frac{\mathbf{Y}\mathbf{b}^*}{\|\mathbf{Y}\mathbf{b}^*\|_2}. \tag{14}$$

When the rank of the nominal signal $\mathbf{Y_L}$ is $d > 1$, the problem $\mathcal{P}^{L_1}$ in (6) can be solved by

$$\begin{aligned}
\mathbf{R}_{L_1} &= \arg \max_{\substack{\mathbf{R}\in\mathbb{R}^{P\times d}\\ \mathbf{R}^T\mathbf{R}=I_d}} \|\mathbf{Y}^T\mathbf{R}\|_1 && (15)\\
&= \arg \max_{\substack{\mathbf{R}\in\mathbb{R}^{P\times d}\\ \mathbf{R}^T\mathbf{R}=I_d}} \max_{\mathbf{B}\in\{\pm1\}^{N\times d}} \operatorname{tr}(\mathbf{R}^T\mathbf{Y}\mathbf{B}) \ \text{ and} && (16)\\
\mathbf{B}_{\text{opt}} &= \max_{\mathbf{B}\in\{\pm1\}^{N\times d}} \|\mathbf{Y}\mathbf{B}\|_* && (17)
\end{aligned}$$

where $\|\cdot\|_*$ stands for nuclear norm. The complexity of solving (17) is $\mathcal{O}(2^{Nd}\min\{P^2d, Pd^2\})$ by the exhaustive search algorithm,[25] $\mathcal{O}(N^{\text{rank}(\mathbf{Y})d-d+1})$ by the polynomial-time algorithm,[25] and $\mathcal{O}(N^3)$ with the SBF algorithm[24,].[26] The SBF algorithm for $d > 1$ starts with an initial binary matrix $\mathbf{B}^{(\text{ini})} \in \{\pm1\}^{N\times d}$ and iteratively produces a sequence of new binary matrices $\mathbf{B}^{(k)}$, $k = 1, 2, ...$, where $\mathbf{B}^{(k+1)}$ at the $(k+1)$-th iteration differs from $\mathbf{B}^{(k)}$ at the kth iteration only in a single bit position. At each iteration we choose to flip the bit that results in the highest increase of $\|\mathbf{Y}\mathbf{B}\|_*$ to reach a (suboptimal) solution of (17), say $\mathbf{B}^*$. The associated L_1-norm projection operator $\mathbf{R}_{L_1}$ can be obtained by performing the following steps:
1) Perform SBF with input $\mathbf{B}^{(\text{ini})}$ to obtain $\mathbf{B}^*$.
2) Perform singular value decomposition (SVD) on $\mathbf{Y}\mathbf{B}^* = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$.
3) Return $\mathbf{R}_{L_1} = \mathbf{U}_{:,1:d}\mathbf{V}^T$.

By projecting the observed CS measurements $\mathbf{Y}$ onto $\mathbf{R}_{L_1}$, we obtain/estimate the compressed-sensed low-rank component which represents the CS background component

$$\widehat{\mathbf{Y}}_{\mathbf{L}} = \mathbf{R}_{L_1}\mathbf{R}_{L_1}^T\mathbf{Y}. \tag{18}$$

2.3 Total-variation Minimization for Pixel-domain Background Reconstruction and Foreground Extraction

Next, the background scene can be reconstructed by performing CS recovery on the columns of $\widehat{\mathbf{Y}}_{\mathbf{L}}$, $\widehat{\mathbf{y}}_{\mathbf{L},t}$, $t = 1, 2, ..., N$. Under the assumption that images are mostly piece-wise smooth, it is natural to exploit the sparsity of the spatial gradient of video frames[27−33] for pixel-domain background scene reconstruction by total-variation (TV)[*] minimization[23,24] of the form

$$\tilde{\boldsymbol{\ell}}_t = \arg\min_{\boldsymbol{\ell}_t} \mu\text{TV}_{2\text{D}}(\boldsymbol{\ell}_t) + \frac{1}{2}\|\widehat{\mathbf{y}}_{\mathbf{L},t} - \mathbf{\Phi}\boldsymbol{\ell}_t\|_2^2. \tag{19}$$

To extract the sparse moving objects in the foreground, we perform frame-by-frame CS reconstruction in the form of

$$\tilde{\mathbf{x}}_t = \arg\min_{\mathbf{x}_t}[\mu\text{TV}_{2\text{D}}(\mathbf{x}_t) + \frac{1}{2}\|\mathbf{y}_t - \mathbf{\Phi}\mathbf{x}_t\|_2^2]. \tag{20}$$

With the recovered video frames $\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1 \ \ \tilde{\mathbf{x}}_2 \ \ \ldots \tilde{\mathbf{x}}_N] \in \mathbb{R}^{D\times N}$ the sparse foreground can be recovered as

$$\tilde{\mathbf{e}}_t = \tilde{\mathbf{x}}_t - \tilde{\boldsymbol{\ell}}_t \tag{21}$$

for $t = 1, 2, ..., N$.

[*] The definition of total-variation (TV) minimization can be found in prior works[32,].[33]

3. THE PROPOSED CS-DOMAIN L_1-SUBSPACE-UPDATES ALGORITHM

The CS-domain L_1-norm algorithm presented in Section 2 is suitable for offline background and foreground extraction in which a fixed number of N compressed-sensed frames are collected at the monitoring decoder and processed. In this section, we consider the problem of online background tracking where the decoder continuously receives new frames. With on-line updated background scenes, we expect to be able to extract the foreground moving objects in an on-line instantaneous manner.

In the Liu and Pados work,[24] an adaptive CS-L_1-PCA algorithm is proposed in which k background frames that are already identified by processing preceding frames are utilized in processing new frames. In this context,[24] the L_1-norm maximization in (7) or in (15) needs to be carried out for every processing window. Since the problems of solving for $\mathbf{r}_{L_1}$ in (7) or $\mathbf{R}_{L_1}$ in (15) boil down to the problems of solving for $\mathbf{b}_{\text{opt}}$ and $\mathbf{B}_{\text{opt}}$ in (11) and (17), respectively, we propose an online L_1-subspace updating scheme that directly operates on the binary antipodal vector $\mathbf{b}$ or matrix $\mathbf{B}$. Instead of computing a new optimal $\mathbf{b}$ or $\mathbf{B}$ from scratch as in (11) and (17), in this paper we modify (11) and (17) for the new processing window such that $\mathbf{b}_{\text{opt}}$ and $\mathbf{B}_{\text{opt}}$ computed from preceding frames are utilized to speed up processing.

3.1 Problem Statement

We start from N initial CS frames $\mathbf{Y} = [\mathbf{y}_1 \quad \mathbf{y}_2 \quad ... \quad \mathbf{y}_N] \in \mathbb{R}^{P\times N}$. To extract background and foreground, we solve (15) to obtain $\mathbf{R}_{L_1} \in \mathbb{R}^{P\times d}$ and the associated optimal binary matrix $\mathbf{B}_{\text{opt}}$. When a new CS frame $\mathbf{y}_t \in \mathbb{R}^P$ arrives at the decoder, we hope to exploit the information contained in $\mathbf{R}_{L_1}$, $\mathbf{B}_{\text{opt}}$ and $\mathbf{y}_t$ to obtain an updated L_1-subspace $\mathbf{R}_t \in \mathbb{R}^{P\times d}$.

To avoid processing an enlarging data ensemble, the size of the processing window is fixed at N frames. Hence, before including the new frame $\mathbf{y}_t$ into the data ensemble, one old frame needs to be removed from $\mathbf{Y}$. Assume that the i-th frame is removed, $1 \le i \le N$, we denote the remaining $N-1$ frames by $\mathbf{Y}_{/i} = [\mathbf{y}_1, ..., \mathbf{y}_{i-1}, \mathbf{y}_{i+1}, ..., \mathbf{y}_N] \in \mathbb{R}^{P\times(N-1)}$. We then append the new frame $\mathbf{y}_t$ to $\mathbf{Y}_{/i}$ and form the new data ensemble at time t, $\mathbf{Y}_t \triangleq [\mathbf{Y}_{/i} \quad \mathbf{y}_t] \in \mathbb{R}^{P\times N}$. Now, the task is to find an updated L_1-subspace $\mathbf{R}_t \in \mathbb{R}^{P\times d}$.

3.2 Rank-1 CS-domain L_1-subspace Updates

First, we consider the case $d = 1$ in which the problem is to find

$$\mathbf{r}_t = \arg \max_{\substack{\mathbf{r}\in\mathbb{R}^P \\ \|\mathbf{r}\|_2=1}} \left\|\mathbf{Y}_t^T \mathbf{r}\right\|_1 . \tag{22}$$

The associated optimization problem with respect to a binary antipodal vector $\mathbf{b}_t \in \{\pm1\}^N$ is

$$\begin{aligned}
\mathbf{b}_t &= \arg \max_{\mathbf{b}\in\{\pm1\}^N} \mathbf{b}^T \mathbf{Y}_t^T \mathbf{Y}_t \mathbf{b} && (23)\\
&= \arg \max_{\begin{bmatrix}\mathbf{b}_{/i}\\ b_t\end{bmatrix}\in\{\pm1\}^N} [\mathbf{b}_{/i}^T \quad b_t] \begin{bmatrix}\mathbf{Y}_{/i}^T\\ \mathbf{y}_t^T\end{bmatrix} [\mathbf{Y}_{/i} \quad \mathbf{y}_t] \begin{bmatrix}\mathbf{b}_{/i}\\ b_t\end{bmatrix} && (24)\\
&= \arg \max_{\begin{bmatrix}\mathbf{b}_{/i}\\ b_t\end{bmatrix}\in\{\pm1\}^N} \{\mathbf{b}_{/i}^T \mathbf{Y}_{/i}^T \mathbf{Y}_{/i} \mathbf{b}_{/i} + 2 b_t \mathbf{y}_t^T \mathbf{Y}_{/i} \mathbf{b}_{/i} + b_t^2 \mathbf{y}_t^T \mathbf{y}_t\} && (25)
\end{aligned}$$

in which $\mathbf{b}_{/i} \in \{\pm1\}^{N-1}$ is the binary antipodal vector associated with the $N-1$ old frames $\mathbf{Y}_{/i}$ and b_t is the binary antipodal bit associated with the new frame $\mathbf{y}_t$.

As we discussed, we wish to avoid solving (23) from scratch and instead exploit information contained in the initial optimal binary vector $\mathbf{b}_{\text{opt}}$ computed by (11) leading to an approximate solution of the optimal vector $\mathbf{b}_t$. By these considerations, we remove the i-th bit from $\mathbf{b}_{\text{opt}}$ for the initial data ensemble $\mathbf{Y}$ and the i-th frame from matrix $\mathbf{Y}$ following the criterion

$$\text{remove} \quad (\mathbf{y}_i,\, b_i) \quad \text{s.t.} \quad i = \arg \max_{1\le i\le N} \left\|\mathbf{y}_i - \mathbf{r}_{L_1}\mathbf{r}_{L_1}^T \mathbf{y}_i\right\|_2^2, \qquad \mathbf{y}_i = [\mathbf{Y}]_{:,i}, \tag{26}$$

and denote the remaining length-$(N-1)$ antipodal bit vector by $\mathbf{b}_{\text{opt}/i}$. This way, we discard the frame with the maximum error between $\mathbf{y}_i$ and its projection on $\mathbf{r}_{L_1}$. Such a frame is more likely to contain moving objects, therefore its removal shall lead to a better estimate of the new background. If we assume $\mathbf{b}_{\text{opt}/i}$ is a good approximation of the optimal partial binary antipodal vector $\mathbf{b}_{/i}$ in (24), which is associated with the $N-1$ old frames $\mathbf{Y}_{/i}$, then the problem in (25) becomes to find the bit b_t that maximizes $2b_t\mathbf{y}_t^T\mathbf{Y}_{/i}\mathbf{b}_{\text{opt}/i}$. The solution to that is simply

$$\widehat{b}_t = \text{sgn}\{\mathbf{y}_t^T\mathbf{Y}_{/i}\mathbf{b}_{\text{opt}/i}\}. \tag{27}$$

Then, we initialize $\mathbf{b}_t^{(\text{ini})} = [\mathbf{b}_{\text{opt}/i}^T \quad \widehat{b}_t]^T$ and carry out single-bit-flipping until no single bit flip can increase the objective function $\mathbf{b}_t^T\mathbf{Y}_t^T\mathbf{Y}_t\mathbf{b}_t$ obtaining an approximate solution $\mathbf{b}_t^*$ to (23). We observe that the number of iterations required for the SBF algorithm to obtain $\mathbf{b}_t^*$ with initial input $\mathbf{b}_t^{(\text{ini})}$ is much fewer than that required by computing a brand new optimal $\mathbf{b}_t$ in (23). Then, we update the new L_1 principal component $\mathbf{r}_t$ and background vector $\widehat{\mathbf{y}}_{\mathbf{L},t}$ by

$$\mathbf{r}_t = \frac{\mathbf{Y}_t\mathbf{b}_t^*}{\|\mathbf{Y}_t\mathbf{b}_t^*\|_2}, \tag{28}$$

$$\widehat{\mathbf{y}}_{\mathbf{L},t} = \mathbf{r}_t\mathbf{r}_t^T\mathbf{y}_t. \tag{29}$$

When the next frame $\mathbf{y}_{t+1}$ arrives at the decoder, the i-th bit of $\mathbf{b}_t^*$ and the i-th frame of $\mathbf{Y}_t$ are removed following a similar criterion to (26)

$$\text{remove} \qquad (\mathbf{y}_i,\, b_i) \quad \text{s.t.} \quad i = \arg\max_{1\le i\le N} \left\|\mathbf{y}_i - \mathbf{r}_t\mathbf{r}_t^T\mathbf{y}_i\right\|_2^2, \qquad \mathbf{y}_i = [\mathbf{Y}_t]_{:,i}, \tag{30}$$

and we form the new data ensemble $\mathbf{Y}_{t+1} \triangleq [\mathbf{Y}_{t/i} \quad \mathbf{y}_{t+1}]$. To find the new L_1 principal component $\mathbf{r}_{t+1}$ for $\mathbf{Y}_{t+1}$, we solve (23) in which $\mathbf{Y}_t$ is replaced by $\mathbf{Y}_{t+1}$ and $\mathbf{b}_t$ is replaced by $\mathbf{b}_{t+1} = [\mathbf{b}_{t/i}^T \quad b_{t+1}]^T$ leading to the following formulation:

$$\begin{aligned} \mathbf{b}_{t+1} &= \arg\max_{\mathbf{b}\in\{\pm1\}^N} \mathbf{b}^T\mathbf{Y}_{t+1}^T\mathbf{Y}_{t+1}\mathbf{b} & (31)\\ &= \arg\max_{\begin{bmatrix}\mathbf{b}_{t/i}\\ b_{t+1}\end{bmatrix}\in\{\pm1\}^N} [\mathbf{b}_{t/i}^T \quad b_{t+1}] \begin{bmatrix}\mathbf{Y}_{t/i}^T\\ \mathbf{y}_{t+1}^T\end{bmatrix} [\mathbf{Y}_{t/i} \quad \mathbf{y}_{t+1}] \begin{bmatrix}\mathbf{b}_{t/i}\\ b_{t+1}\end{bmatrix} & (32) \end{aligned}$$

where $\mathbf{b}_{t/i} \in \{\pm1\}^{N-1}$ is the binary antipodal vector associated with the $N-1$ old frames $\mathbf{Y}_{t/i}$ and b_{t+1} is the binary antipodal bit associated with the new frame $\mathbf{y}_{t+1}$. Again, our objective is to find an approximate solution to (31). We initialize $\mathbf{b}_{t/i}$ by $\mathbf{b}_{t/i}^*$, that is the suboptimal solution to (23) excluding the i-th bit, b_{t+1} by $\widehat{b}_{t+1} = \text{sgn}\{\mathbf{y}_{t+1}^T\mathbf{Y}_{t/i}\mathbf{b}_{t/i}^*\}$ and $\mathbf{b}_{t+1}$ by $\mathbf{b}_{t+1}^{(\text{ini})} = [\mathbf{b^*}_{t/i}^T \quad \widehat{b}_{t+1}]^T$ and perform SBF with $\mathbf{Y}_{t+1}$ and $\mathbf{b}_{t+1}^{(\text{ini})}$ as the input to obtain $\mathbf{b}_{t+1}^*$. Finally, we update the new L_1 principal component $\mathbf{r}_{t+1}$ and background vector $\widehat{\mathbf{y}}_{\mathbf{L},t+1}$ by

$$\mathbf{r}_{t+1} = \frac{\mathbf{Y}_{t+1}\mathbf{b}_{t+1}^*}{\left\|\mathbf{Y}_{t+1}\mathbf{b}_{t+1}^*\right\|_2}, \tag{33}$$

$$\widehat{\mathbf{y}}_{\mathbf{L},t+1} = \mathbf{r}_{t+1}\mathbf{r}_{t+1}^T\mathbf{y}_{t+1}. \tag{34}$$

3.3 Rank-d $(d > 1)$ CS-domain L_1-subspace Updates

All these considerations can be easily extended to the multiple principal-components case. Considering the same initially collected data ensemble $\mathbf{Y}$ of the previous section, the initial L_1 principal components $\mathbf{R}_{L_1} \in \mathbb{R}^{P\times d}$ and its associated binary antipodal matrix $\mathbf{B}_{\text{opt}} \in \{\pm1\}^{N\times d}$ can be found via (15)-(17). The compressed-sensed background (low-rank) component is given by

$$\widehat{\mathbf{Y}}_{\mathbf{L}} = \mathbf{R}_{L_1}\mathbf{R}_{L_1}^T\mathbf{Y}. \tag{35}$$

When a new sample $\mathbf{y}_t$ arrives, following previous considerations on the moving window, the criterion to remove frames is

$$\text{remove} \qquad (\mathbf{y}_i,\, \mathbf{b}_i) \quad \text{s.t.} \quad i = \arg\max_{1 \leq i \leq N} \left\| \mathbf{y}_i - \mathbf{R}_{L_1}\mathbf{R}_{L_1}^T \mathbf{y}_i \right\|_2^2, \qquad \mathbf{y}_i = [\mathbf{Y}]_{:,i}, \tag{36}$$

where $\mathbf{b}_i^T \in \{\pm 1\}^{1\times d}$ is the i-th row of matrix $\mathbf{B}_{\text{opt}}$. We set the new data matrix to $\mathbf{Y}_t = [\mathbf{Y}_{/i} \quad \mathbf{y}_t]$. Then, we solve the following problem to find the new L_1-subspace $\mathbf{R}_t$:

$$\mathbf{R}_t = \arg\max_{\substack{\mathbf{R}\in\mathbb{R}^{P\times d} \\ \mathbf{R}^T\mathbf{R}=I_d}} \|\mathbf{Y}_t^T \mathbf{R}\|_1 \tag{37}$$

$$= \arg\max_{\substack{\mathbf{R}\in\mathbb{R}^{P\times d} \\ \mathbf{R}^T\mathbf{R}=I_d}} \max_{\mathbf{B}\in\{\pm1\}^{N\times d}} \operatorname{tr}(\mathbf{R}^T \mathbf{Y}_t \mathbf{B}), \text{ and} \tag{38}$$

$$\mathbf{B}_t = \arg\max_{\mathbf{B}\in\{\pm1\}^{N\times d}} \|\mathbf{Y}_t \mathbf{B}\|_* \tag{39}$$

$$= \arg\max_{\left[\begin{array}{c}\mathbf{B}_{/i} \\ \mathbf{b}_t^T\end{array}\right]\in\{\pm1\}^{N\times d}} \left\| [\mathbf{Y}_{/i} \quad \mathbf{y}_t] \left[\begin{array}{c}\mathbf{B}_{/i} \\ \mathbf{b}_t^T\end{array}\right] \right\|_*. \tag{40}$$

Again we wish to avoid solving (39) from scratch. We assume that $\mathbf{B}_{\text{opt}/i}$ is a good approximation of the partial binary matrix $\mathbf{B}_{/i}$ in (40) and seek the vector $\widehat{\mathbf{b}}_t$ such that

$$\widehat{\mathbf{b}}_t = \arg\max_{\mathbf{b}_t\in\{\pm1\}^d} \left\| \mathbf{Y}_t \left[\begin{array}{c}\mathbf{B}_{\text{opt}/i} \\ \mathbf{b}_t^T\end{array}\right] \right\|_*, \tag{41}$$

which can be obtained by exhaustive search over the 2^d combinations $\{\pm1\}^d$. Then, we set $\mathbf{B}_t^{(\text{ini})} = \left[\begin{array}{c}\mathbf{B}_{\text{opt}/i} \\ \widehat{\mathbf{b}}_t^T\end{array}\right]$ and perform single-bit-flipping with input $\mathbf{Y}_t$ and $\mathbf{B}_t^{(\text{ini})}$ until no single bit flip in $\mathbf{B}_t$ can increase the value of $\|\mathbf{Y}_t\mathbf{B}_t\|_*$ leading to an approximate solution of (39), say $\mathbf{B}_t^*$. Again, in this case the number of iterations needed for SBF to converge is much fewer than that required to compute a new optimal $\mathbf{B}_t$ in (39). Finally, we update subspace $\mathbf{R}_t$ and the background vector $\widehat{\mathbf{y}}_{\mathbf{L},t}$ by

$$\mathbf{R}_t = \mathbf{U}_t \mathbf{V}_t^T, \tag{42}$$

$$\widehat{\mathbf{y}}_{\mathbf{L},t} = \mathbf{R}_t \mathbf{R}_t^T \mathbf{y}_t, \tag{43}$$

where $\mathbf{U}_t$ and $\mathbf{V}_t$ are the $P \times d$ and $N \times d$ matrices that consists of the d dominant-singular-value, left and right respectively, singular vectors of $\mathbf{Y}_t\mathbf{B}_t^*$.

When a new sample $\mathbf{y}_{t+1}$ arrives at the decoder, we perform

1) Remove $(\mathbf{y}_i,\, \mathbf{b}_i)$ s. t. $i = \arg\max_{1\leq i\leq N} \left\|\mathbf{y}_i - \mathbf{R}_t\mathbf{R}_t^T\mathbf{y}_i\right\|_2^2$, $\mathbf{y}_i = [\mathbf{Y}_t]_{:,i}$.

2) Set $\mathbf{Y}_{t+1} \triangleq [\mathbf{Y}_{t/i} \quad \mathbf{y}_{t+1}]$.

3) Solve

$$\mathbf{B}_{t+1} = \arg\max_{\mathbf{B}\in\{\pm1\}^{N\times d}} \|\mathbf{Y}_{t+1}\mathbf{B}\|_*, \tag{44}$$

$$= \arg\max_{\left[\begin{array}{c}\mathbf{B}_{t/i} \\ \mathbf{b}_{t+1}^T\end{array}\right]\in\{\pm1\}^{N\times d}} \left\| [\mathbf{Y}_{t/i} \quad \mathbf{y}_{t+1}] \left[\begin{array}{c}\mathbf{B}_{t/i} \\ \mathbf{b}_{t+1}^T\end{array}\right] \right\|_*, \tag{45}$$

in which $\mathbf{B}_{t/i}$ is approximated by $\mathbf{B}^*_{t/i}$ and $\mathbf{b}_{t+1}$ is initialized as $\widehat{\mathbf{b}}_{t+1} = \arg \max\limits_{\mathbf{b}_{t+1} \in \{\pm 1\}^d} \left\| \mathbf{Y}_{t+1} \begin{bmatrix} \mathbf{B}^*_{t/i} \\ \mathbf{b}^T_{t+1} \end{bmatrix} \right\|_*$ leading to the initialization $\mathbf{B}^{(\text{ini})}_{t+1} = \begin{bmatrix} \mathbf{B}^*_{t/i} \\ \widehat{\mathbf{b}}^T_{t+1} \end{bmatrix}$.

4) Perform SBF with input $\mathbf{Y}_{t+1}$ and $\mathbf{B}^{(\text{ini})}_{t+1}$ until it converges to $\mathbf{B}^*_{t+1}$.

5) Compute

$$\mathbf{R}_{t+1} = \mathbf{U}_{t+1}\mathbf{V}^T_{t+1}, \tag{46}$$

$$\widehat{\mathbf{y}}_{\mathbf{L},t+1} = \mathbf{R}_{t+1}\mathbf{R}^T_{t+1}\mathbf{y}_{t+1}, \tag{47}$$

where $\mathbf{U}_{t+1}$ and $\mathbf{V}_{t+1}$ are the $P \times d$ and $N \times d$ matrices that consists of the d dominant-singular-value, left and right respectively, singular vectors of $\mathbf{Y}_{t+1}\mathbf{B}^*_{t+1}$.

3.4 Pixel-domain background reconstruction and foreground extraction

For each new frame $\mathbf{y}_t$, the background scene can be reconstructed by performing CS recovery on $\widehat{\mathbf{y}}_{\mathbf{L},t}$ with the procedure introduced in Section 2.3, followed by foreground extraction.

4. EXPERIMENTAL RESULTS

We demonstrate the effectiveness of the proposed adaptive L_1-PCA approach on two test surveillance video sequences, *PETS2001* and *Airport*. For each video sequence, a total number of $N_{\text{tot}} = 200$ frames are selected to form a video volume. Each frame is compressed-sensed independently using the same randomly permuted partial Walsh-Hadamard matrix. The number of CS measurements per frame is 37.5% of the total number of pixels in the video frame. We compare the proposed adaptive L_1-PCA method with non-adaptive L_1-PCA and adaptive L_2-PCA. For the adaptive L_1 and L_2 approaches, we set $N_{\text{tot}} = N + N_{\text{new}}$ where $N = 10$ is the number of frames for initialization and $N_{new} = 190$ is the number of new arriving frames. The non-adaptive L_1-PCA scheme operates on non-overlapping windows of $N = 10$ frames and computes $\frac{N_{tot}}{N} = 20$ subspaces.

4.1 Clean Datasets

First, we carry out experiments on CS measurements that are "clean" (not corrupted by outliers). Background tracking and foreground extraction results are shown in Figs. 2-5. For all experiments, 5 of the N_{new} frames are shown. The frame indices are indicated at the bottom of the figures.

In all experiments, the proposed online CS-L_1-PCA algorithm outperforms its non-adaptive counterpart. While the perceptual quality of the proposed adaptive CS-L_1-PCA is similar to adaptive L_2-PCA, we emphasize that the processing time for adaptive L_2-PCA is much longer than the proposed adaptive L_1-PCA scheme due to the required SVD computation for every L_2-subspace update.

4.2 Corrupted Data Sets

In this section, we demonstrate how the algorithms react to corrupted CS measurements. For both test sequences, 75% of the CS measurements of 50% of the frames (randomly selected) are corrupted by outliers[†].

Figs. 6 and 7 present the background and foreground recovery results for the *PETS2001* sequence. $d = 1$ and $d = 3$ principal components are considered for all algorithms in comparison. When $d = 1$, non-adaptive CS-L_1-PCA and adaptive CS-L_2-PCA are unable to recover the background and foreground scenes (Fig. 6 (ii),(vi) and Fig. 7 (ii),(vi)). On the contrary, the proposed adaptive CS-L_1-PCA procedure is able to recover the background and reveal the foreground moving objects after a few new frames ($t \geq 38$) (Fig. 6 (iv) and Fig. 7 (iv)).

[†] If the i-th CS measurement of the t-th frame, $\mathbf{y}_t(i)$, is selected to be corrupted by an outlier, then in the experiments $\mathbf{y}_t(i)$ is replaced by $\alpha \max\limits_{1 \leq t \leq N_{tot}} |\mathbf{y}_t(i)|$ ($\alpha = -3.5$ herein).

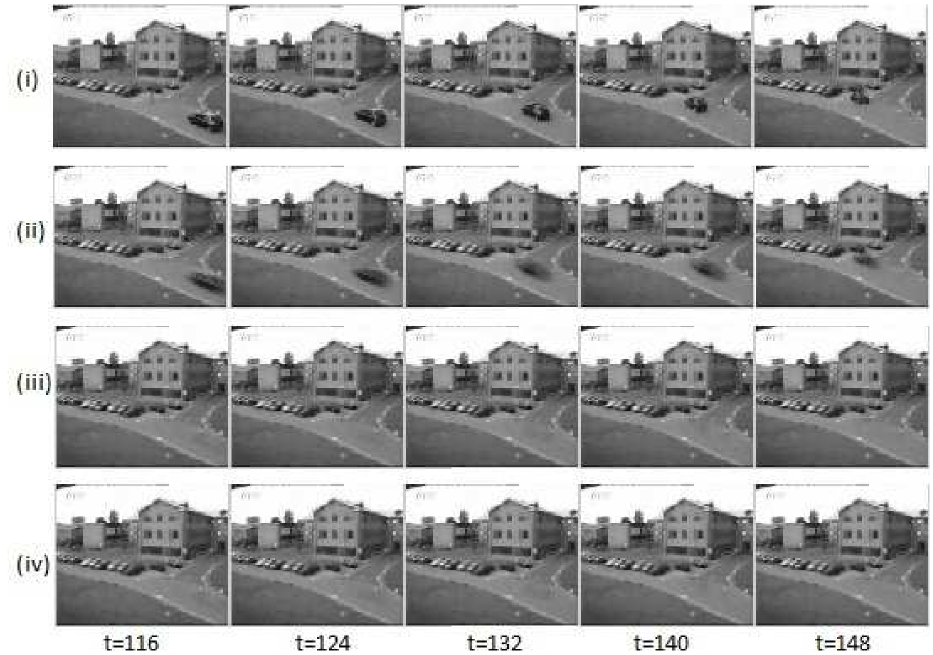

Figure 2. *PETS2001* sequence ("clean" CS measurements): (i) original frames; and reconstructed **background** by (ii) non-adaptive L_1-PCA, (iii) proposed adaptive L_1-PCA, (iv) adaptive L_2-PCA updates.

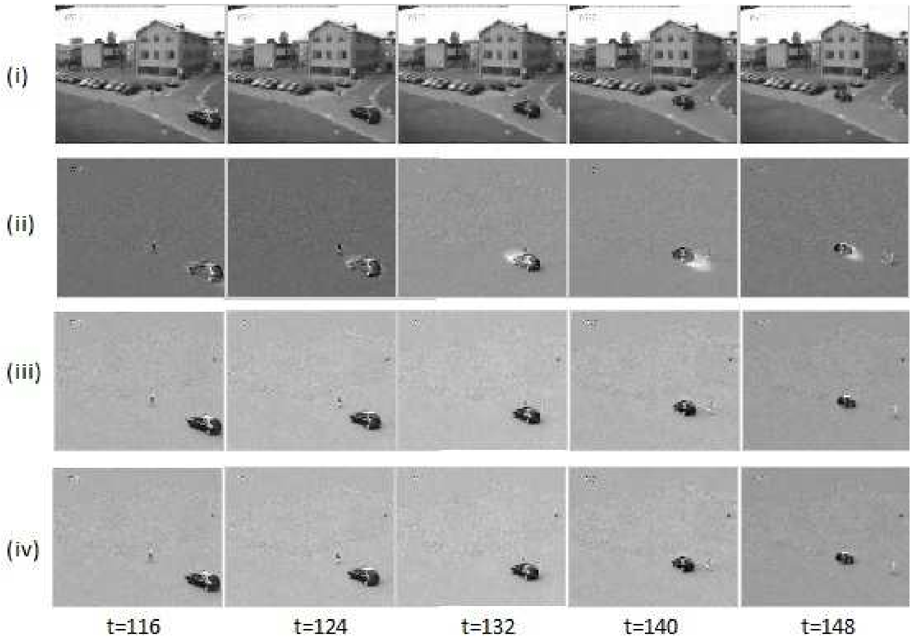

Figure 3. *PETS2001* sequence ("clean" CS measurements): (i) original frames; and extracted **foreground** by (ii) non-adaptive L_1-PCA, (iii) proposed adaptive L_1-PCA, (iv) adaptive L_2-PCA updates.

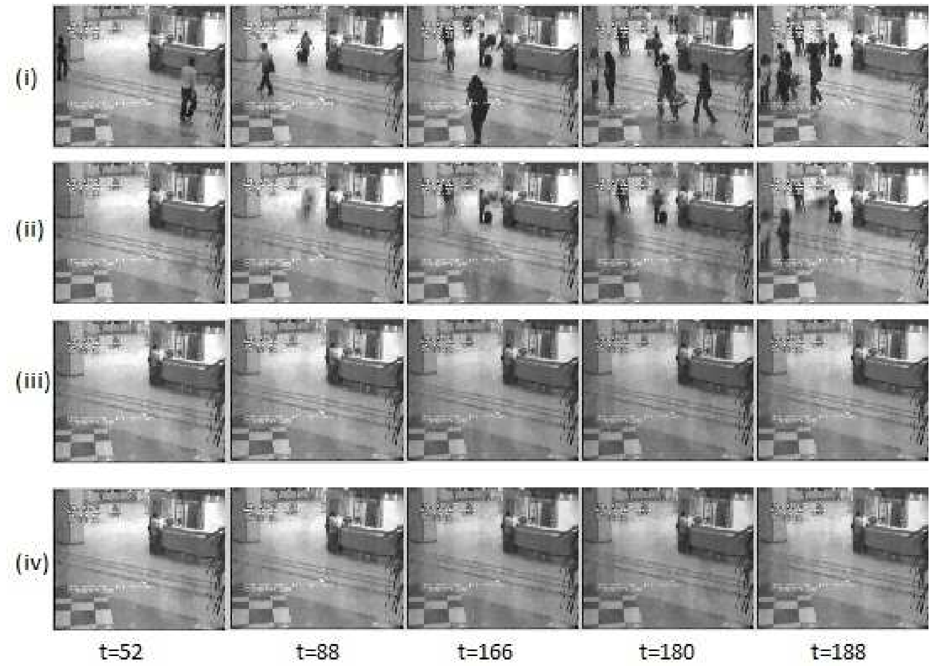

Figure 4. *Airport* sequence ("clean" CS measurements): (i) original frames; and reconstructed **background** by (ii) non-adaptive L_1-PCA, (iii) proposed adaptive L_1-PCA, (iv) adaptive L_2-PCA updates.

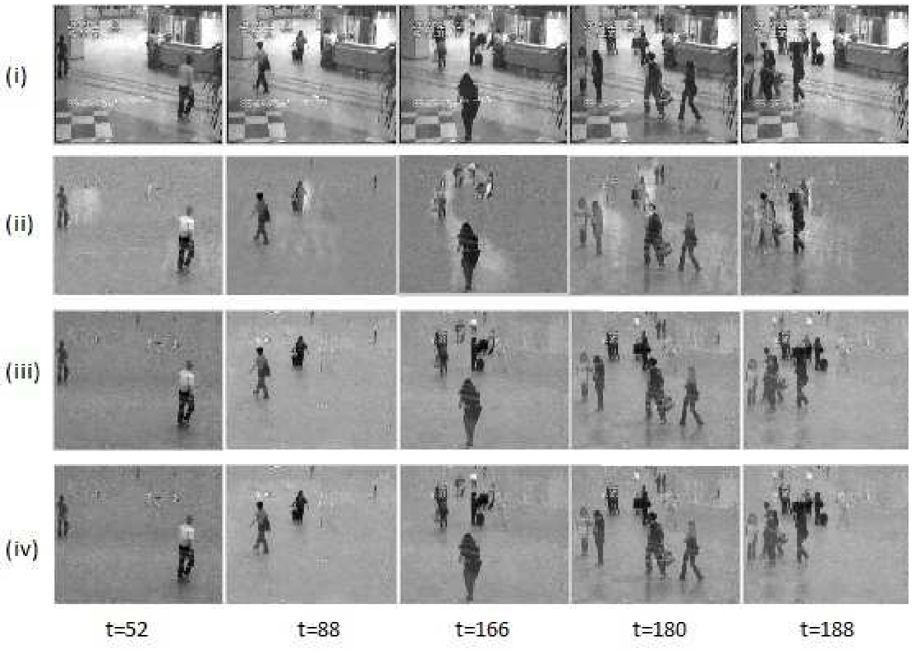

Figure 5. *Airport* sequence ("clean" CS measurements): (i) original frames; and extracted **foreground** by (ii) non-adaptive L_1-PCA, (iii) proposed adaptive L_1-PCA, (iv) adaptive L_2-PCA updates.

When a larger number of principal components is used, $d = 3$, it is expected that the performance of the algorithms is improved. The reason for this is because the presence of outliers modifies/increases the effective SVD rank of the low-rank background scenes. Still, the recovered background and foreground scenes by non-adaptive CS-L_1-PCA (Fig. 6 (iii) and Fig. 7 (iii)) are inaccurate and adaptive CS-L_2-PCA is only able to obtain good results when $t \geq 18$ (Fig. 6 (vii) and Fig. 7 (vii)), which indicates possible detection failures of important foreground objects. On the other hand, the proposed adaptive CS-L_1-PCA procedure achieves good reconstruction quality as early as $t = 14$ (Fig. 6 (v) and Fig. 7 (v)).

The same experiments are carried out for the *Airport* sequence and similar conclusions can be drawn from Figs. 8 and 9.

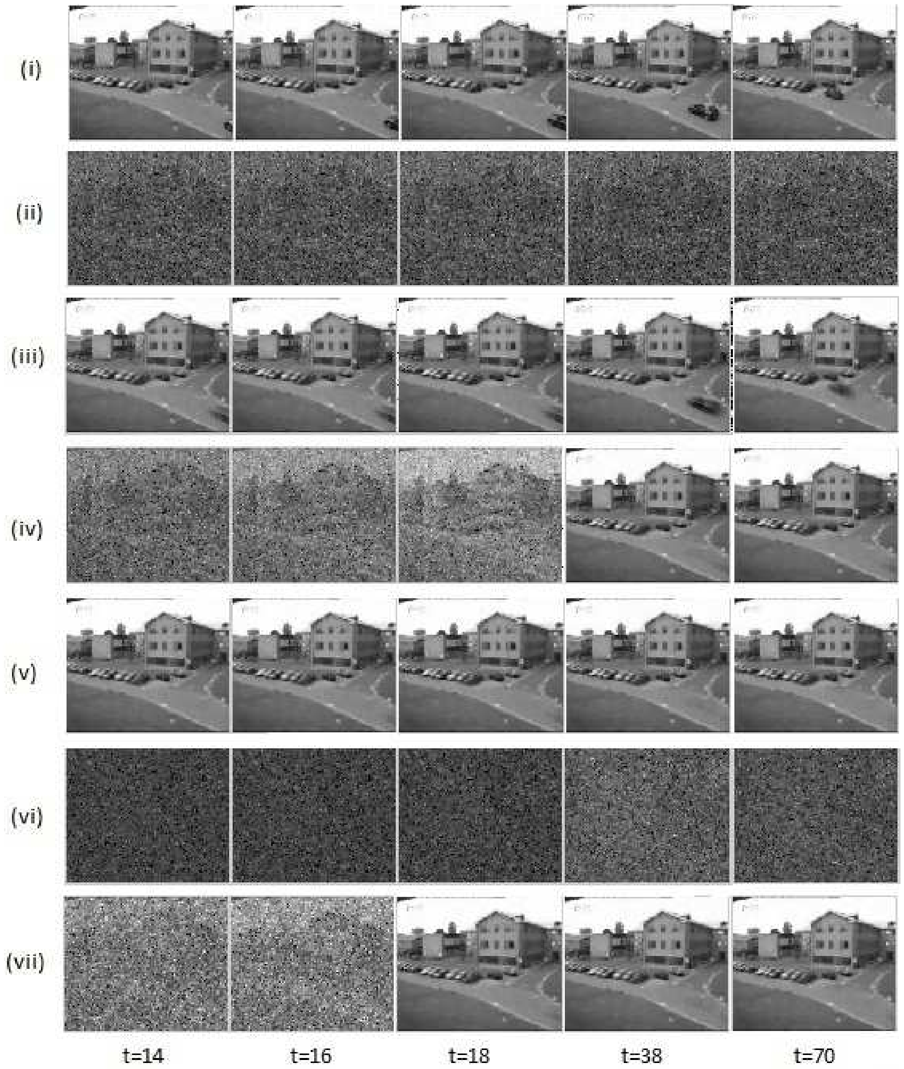

Figure 6. *PEST2001* sequence ("corrupted" CS measurements): (i) original frames; and reconstructed **background** by (ii) non-adaptive L_1-PCA ($d = 1$), (iii) non-adaptive L_1-PCA ($d = 3$), (iv) adaptive L_1-PCA ($d = 1$), (v) adaptive L_1-PCA ($d = 3$), (vi) adaptive L_2-PCA ($d = 1$), (vii) adaptive L_2-PCA ($d = 3$).

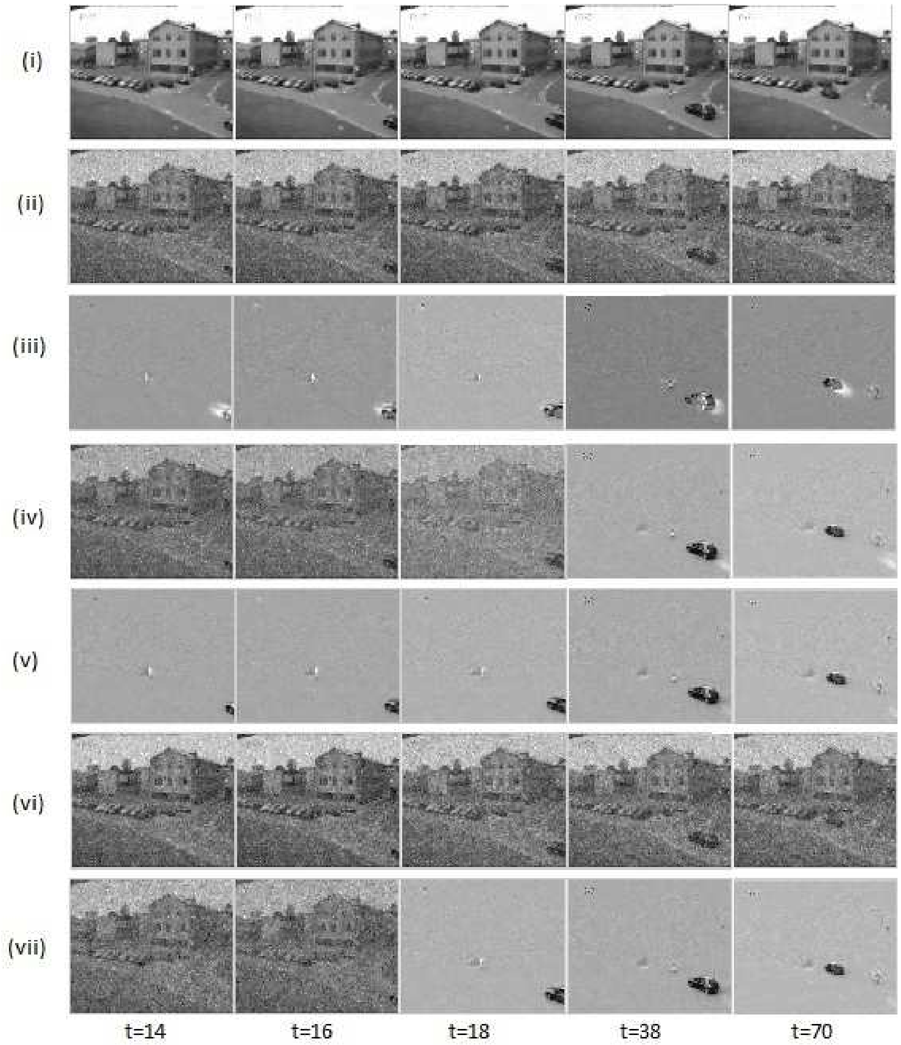

Figure 7. *PEST2001* sequence ("corrupted" CS measurements): (i) original frames; and extracted **foreground** by (ii) non-adaptive L_1-PCA ($d = 1$), (iii) non-adaptive L_1-PCA ($d = 3$), (iv) adaptive L_1-PCA ($d = 1$), (v) adaptive L_1-PCA ($d = 3$), (vi) adaptive L_2-PCA ($d = 1$), (vii) adaptive L_2-PCA ($d = 3$).

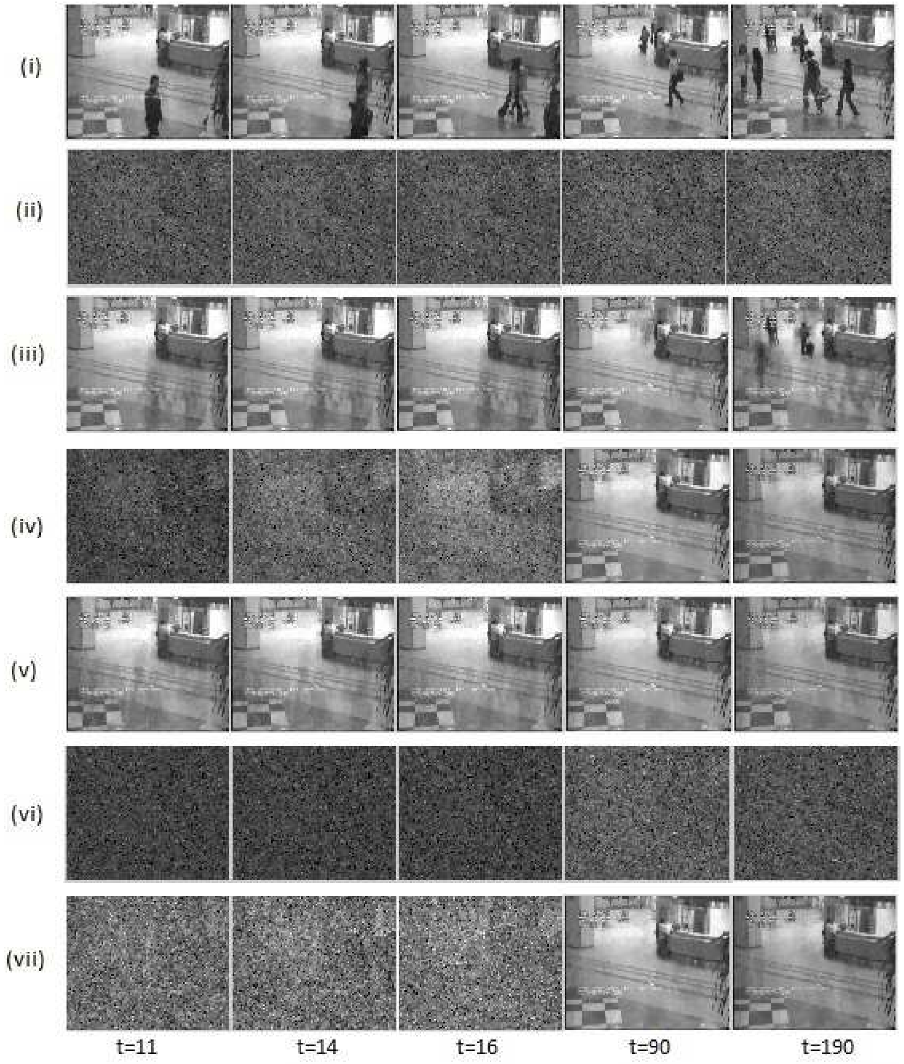

Figure 8. *Airport* sequence ("corrupted" CS measurements): (i) original frames; and reconstructed **background** by (ii) non-adaptive L_1-PCA ($d = 1$), (iii) non-adaptive L_1-PCA ($d = 3$), (iv) adaptive L_1-PCA ($d = 1$), (v) adaptive L_1-PCA ($d = 3$), (vi) adaptive L_2-PCA ($d = 1$), (vii) adaptive L_2-PCA ($d = 3$).

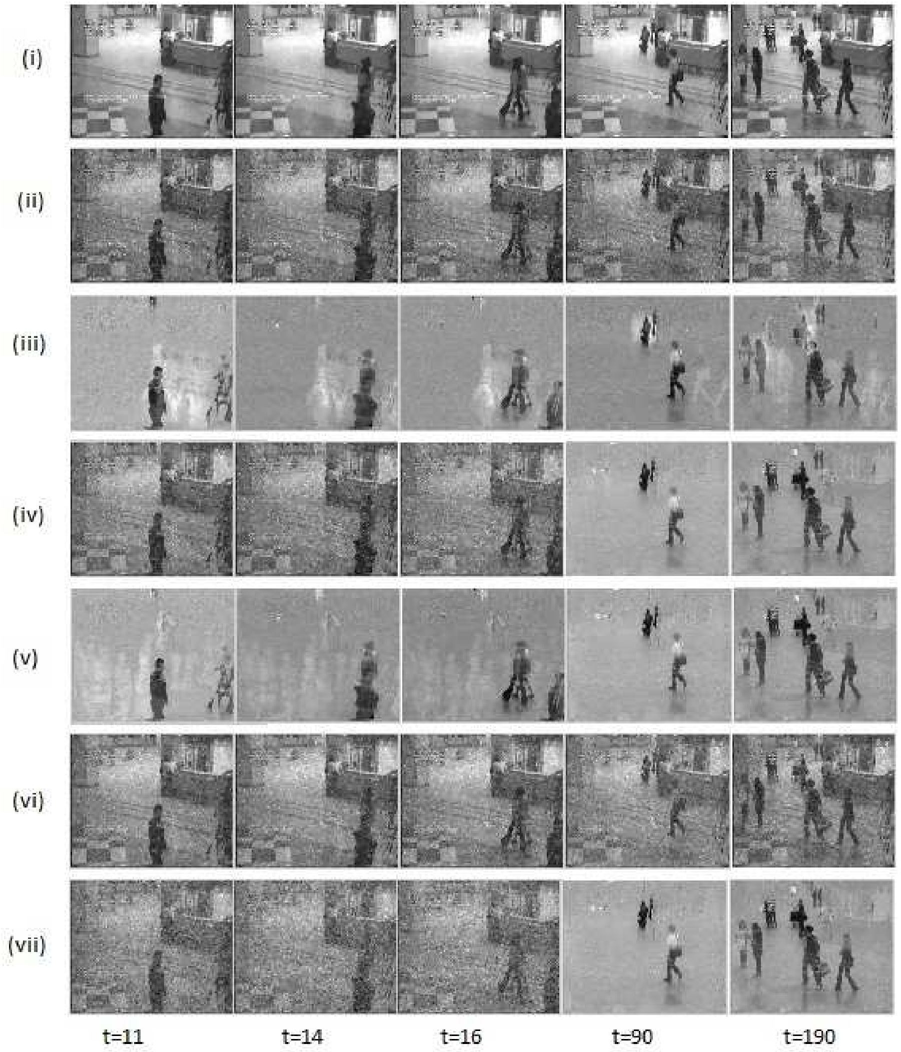

Figure 9. *Airport* sequence ("corrupted" CS measurements): (i) original frames; and extracted **foreground** by (ii) non-adaptive L_1-PCA ($d = 1$), (iii) non-adaptive L_1-PCA ($d = 3$), (iv) adaptive L_1-PCA ($d = 1$), (v) adaptive L_1-PCA ($d = 3$), (vi) adaptive L_2-PCA ($d = 1$), (vii) adaptive L_2-PCA ($d = 3$).

5. CONCLUSIONS

We proposed an adaptive (online) compressed-sensed-domain L_1-subspace update algorithm for compressed-sensed surveillance video processing. For each new video frame arriving at the decoder, the algorithm updates the low-rank subspace for background scene representation via CS-domain L_1-PCA. Background reconstruction is then performed for each new video frame by projecting the CS measurement vector onto the updated L_1 subspace followed by regular CS image recovery (for example, total-variation minimization). Experiments demonstrate that when the received CS measurements are corrupted by outliers, the proposed adaptive CS-L_1-PCA algorithm offers significantly better performance than its L_2-norm counterpart and the non-adaptive CS-L_1-PCA approach. In addition, in clean CS data operation, the proposed adaptive CS-L_1-PCA method requires less processing time for subspace update compared to the adaptive L_2-PCA method. For future studies, more advanced algorithms can be developed under the CS-L_1-PCA framework to deal with more challenging situations in video surveillance such as dynamic background, removing shadows cast by objects to more accurately describe the object shape, and solving the camouflage problem.

REFERENCES

1. Y. Zheng and L. Fan, "Moving object detection based on running average background and temporal difference," in *Proc. IEEE Int. Conf. Intell. Syst. Knowl. Eng. (ISKE)*, Hangzhou, China, Nov. 2010, pp. 270-272.
2. Q. Zhou and J. Aggarwal, "Tracking and classifying moving objects from video," in *Proc. IEEE Int. Workshop on Performance Eval. of Tracking and Surveillance (PETS)*, Kauai, Hawaii, Dec. 2001, pp. 1-8.
3. C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: real-time tracking of the human body," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, pp. 780-785, July 1997.
4. C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Fort Collins, Colorado, June 1999, pp. 246-252.
5. M. Haque, M. Murshed, and M. Paul, "Improved Gaussian mixtures for robust object detection by adaptive multi-background generation," in *Proc. Intern. Conf. Pattern Recog. (ICPR)*, Tampa, FL, Dec. 2008, pp. 1-4.
6. A. Elgammal, D. Harwood, and L. S. Davis, "Nonparametric model for background subtraction," *in Proc. 6th European Conf. Computer Vision (ECCV)-Part II*, Dublin, Ireland, June 2000, pp. 751-767.
7. D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, pp. 603-619, May 2002.
8. M. Piccardi, T. Jan, "Mean-shift background image modeling," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Singapore, Oct. 2004, pp. 3399-3402.
9. Y. Liu, H. Yao, W. Gao, X. Chen, and D. Zhao, "Nonparametric background generation," *J. Vis. Commun. Image Represent.*, vol. 18, pp. 253-263, June 2007.
10. B. Han, D. Comaniciu, and L. S. Davis, "Sequential kernel density approximation through model propagation: applications to background modeling," in *Proc. Asian Conf. Computer Vision*, Jeju, Korea, Jan. 2004, pp. 1-7.
11. B. Chen and S. Huang, "An advanced moving object detection algorithm for automatic traffic monitoring in real-world limited bandwidth networks," *IEEE Trans. Multimedia*, vol. 16, pp. 837-847, Apr. 2014.
12. J. Seo and S. D. Kim, "Recursive on-line $(2D)^2$PCA and its application to long-term background subtraction," *IEEE Trans. Multimedia*, vol. 16, pp. 2333-2344, Dec. 2014.
13. Q. Ke and T. Kanade, "Robust subspace computation using L_1 norm," Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-CS-03-172, Aug. 2003, [Online]. Available: http://ra.adm.cs.cmu.edu/anon/usr0/ftp/usr/anon/2003/CMU-CS-03-172.pdf.
14. Q. Ke and T. Kanade, "Robust norm factorization in the presence of outliers and missing data by alternative convex programming," in *Proc. IEEE Conf. Comput. Vision Pattern Recog. (CVPR)*, San Diego, CA, Jun. 2005, pp. 739-746.
15. A. Eriksson and A. v. d. Hengel, "Efficient computation of robust low-rank matrix approximations in the presence of missing data using the L_1 norm," in *Proc. IEEE Conf. Comput. Vision Pattern Recog. (CVPR)*, San Francisco, CA, Jun. 2010, pp. 771-778.

16. L. Yu, M. Zhang, and C. Ding, "An efficient algorithm for L_1-norm principal component analysis," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process. (ICASSP)*, Kyoto, Japan, Mar. 2012, pp. 1377-1380.

17. N. Guan, D. Tao, Z. Luo, and J. S. Taylor, "MahNMF: Manhattan nonnegative matrix factorization," *CoRR*, Jul. 2012, [Online]. Available: http://arxiv.org/abs/1207.3438.

18. E. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM (JACM)*, vol. 58, pp. 1-37, May 2011.

19. X. Zhou, C. Yang, and W. Yu, "Moving object detection by detecting contiguous outliers in the low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, pp. 597-610, Mar. 2013.

20. H. Jiang, W. Deng, and Z. Shen, "Surveillance video processing using compressive sensing," *Inverse Problems and Imaging*, vol. 6, pp. 201-214, 2012.

21. H. Ganapathy, D. A. Pados, and G. N. Karystinos, "New bounds and optimal binary signature sets - Part I: Periodic total squared correlation," *IEEE Trans. Comm.*, vol. 59, pp. 1123-1132, Apr. 2011.

22. H. Ganapathy, D. A. Pados, and G. N. Karystinos, "New bounds and optimal binary signature sets - Part II: Aperiodic total squared correlation," *IEEE Trans. Comm.*, vol. 59, pp. 1411-1420, May 2011.

23. Y. Liu and D. A. Pados, "Compressed-sensed-domain L_1-PCA video surveillance," *in Proc. SPIE Compressive Sensing Conf., SPIE Defense, Security, and Sensing*, Baltimore, MD, April 2015, pp. 1-10.

24. Y. Liu and D. A. Pados, "Compressed-sensed-domain L_1-PCA video surveillance," *IEEE Trans. Multimedia*, vol. 18, no. 3, pp. 351-363, Mar. 2016.

25. P. P. Markopoulos, G. N. Karystinos, and D. A. Pados, "Optimal algorithms for L_1-subspace signal processing," *IEEE Trans. Signal Process.*, vol. 62, pp. 5046-5058, Oct. 2014.

26. S. Kundu, P. P. Markopoulos and D. A. Pados, "Fast computation of the L_1-principal component of real-valued data," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc. (ICASSP)*, Florence, Italy, May 2014, pp. 8028-8032.

27. E. Candès and J. Romberg, "ℓ1-magic: Recovery of sparse signals via convex programming," www.acm.caltech.edu/l1magic/downloads/l1magic.pdf.

28. M. Lustig, D. Donoho, and J. M. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magn. Reson. Med.*, vol. 6, pp. 1182-1195, Dec. 2007.

29. S. Ma, W. Yin, Y. Zhang, and A. Chakraborty, "An efficient algorithm for compressed MR imaging using total variation and wavelets," in *Proc. IEEE Conf. Comp. Vis. and Pattern Recogn. (CVPR)*, Anchorage, Alaska, Jun. 2008, pp. 1-8.

30. C. Li, "An efficient algorithm for total variation regularization with applications to the single pixel camera and compressive sensing," M.S. thesis, Dept. Comp. Appl. Math., Rice Univ., Houston, TX, USA, 2009.

31. M. R. Dadkhah, S. Shirani, and M. J. Deen, "Compressive sensing with modified total variation minimization algorithm," in *Proc. IEEE Int. Conf. Acoust, Speech, and Signal Proc. (ICASSP)*, Dallas, TX, Mar. 2010, pp. 1310-1313.

32. Y. Liu and D. A. Pados, "Decoding of framewise compressed-sensed video via interframe total variation minimization," *SPIE J. Electron. Imaging, Special Issue on Compressive Sensing for Imaging*, vol. 22, pp. 1-8, Apr.-June 2013.

33. Y. Liu and D. A. Pados, "Rate-adaptive compressive video acquisition with sliding-window total-variation-minimization reconstruction," in *Proc. SPIE Compressive Sensing Conf., SPIE Defense, Security, and Sensing*, Baltimore, MD, May 2013, pp. 1-13.