

Joint compressive video coding and analysis

Michele Cossalter, Giuseppe Valenzise *Student Member, IEEE*, Marco Tagliasacchi *Member, IEEE*, Stefano Tubaro *Member, IEEE*

Abstract

Traditionally, video acquisition, coding and analysis have been designed and optimized as independent tasks. This has a negative impact in terms of consumed resources, as most of the raw information captured by conventional acquisition devices is discarded in the coding phase, while the analysis step only requires a few descriptors of salient video characteristics. Recent Compressive Sensing literature has partially broken this paradigm by proposing to integrate sensing and coding in a unified architecture composed by a light encoder and a more complex decoder, which exploits sparsity of the underlying signal for efficient recovery. However, a clear understanding of how to embed video analysis in this scheme is still missing. In this paper, we propose a joint compressive video coding and analysis scheme and, as a specific application example, we consider the problem of object tracking in video sequences. We show that, weaving together compressive sensing and the information computed by the analysis module, the bit-rate required to perform reconstruction and tracking of the foreground objects can be considerably reduced, with respect to a conventional disjoint approach that postpones the analysis after the video signal is recovered in the pixel domain. These findings suggest that considerable gains in performance can be potentially obtained in video analysis applications, provided that a joint analysis-aware design of acquisition, coding and signal recovery is carried out.

Index Terms

Compressive sensing, video coding, video analysis.

I. INTRODUCTION

When acquiring a digital video stream, the goal may not be necessarily to display it with the best possible visual quality to the end-user. In many cases, such as in video surveillance applications, the

The authors are with Dipartimento di Elettronica e Informazione, Politecnico di Milano, P.za Leonardo da Vinci, 32 20133 - Milano, Italy - Ph. +39-02-2399-7624 - FAX: +39-02-2399-7321 - E-mail: cossalter@elet.polimi.it, marco.tagliasacchi@polimi.it, valenzise@elet.polimi.it. This work has been partially sponsored by the EU under Visnet II Network of Excellence. Part of the material presented in this paper has been accepted for publication in the 6th IEEE International Conference on Automatic Video and Signal-Based Surveillance, Genoa, Italy, September 2009.

acquired video could be automatically processed in order to perform further analysis tasks and extract relevant information. Most of these high-level activities entail the summarization of salient aspects of the video through a small set of semantically relevant features, such as the dimension and/or the speed of objects moving in the scene, etc. Once this aggregated information is computed, the analysis task can carry on, while all the additional low-level information contained in the raw video stream (i.e. most part of the acquired signal) is discarded. Conventional video analysis approaches are based on this sample-compress-and-analyze strategy, with the three activities being designed and optimized separately one from each other. This can be inefficient, since both acquisition and coding are carried out on the entire signal, while most of their results are discarded in the compression and in the analysis processes, with a noticeable waste of bandwidth and storage resources. Moreover, in some acquisition devices, such as medical scanners or imaging systems working at wavelengths where cheap CMOS or CCD sensors are ineffective, this approach may be unfavorable especially from the point of view of the costs of acquisition devices.

Recently, a breakthrough in this matter has been provided by the Compressive Sensing (CS) theory [1], [2], [3], which asserts that it is possible to blend together the sampling and coding stages to acquire certain signals and images (namely, the ones that can be represented by a sparse set of coefficients in a proper basis) directly in a compressed form, using far fewer samples or measurements than traditional methods use. In a CS acquisition architecture, the computational effort is asymmetrically distributed between a light encoder, which collects a small number of linear measurements of the signal by correlating it against a set of random vectors, and a more complex decoder, where the signal is reconstructed by solving a convex optimization problem. The single-pixel camera of [4] has been the first prototype of a CS-based acquisition device following this paradigm. This hardware is able to optically compute incoherent measurements using a micro-controlled mirror (MCM) array driven by pseudorandom bases and a single photodiode optical sensor. From the acquired data, an image representation in the pixel domain can be recovered exploiting CS reconstruction techniques. CS ideas can also be applied to low resolution conventional devices in order to get super-resolved pictures [5]: in this case, the linear projections are obtained as the output of the convolution between the acquired image and a random filter, while the reconstruction is performed following the standard CS procedure. The performance of CS methods is strongly influenced by the ability of finding a sparse representation of the signal in an appropriate basis. This is especially true for the case of video, where a good sparsification inevitably involves a proper handling of the temporal correlation between frames. In [6], it is proposed to exploit this correlation by using a 3D wavelet transform across spatial and temporal dimensions. The main drawback of this solution is its high complexity, since it requires the whole sequence to be acquired and then jointly encoded. An alternative approach is presented in [7],

where the authors develop a system that splits each frame in blocks and applies CS to sparse blocks in order to reduce the required sampling rate. An optimal way to sparsify a video across the temporal dimension consists in computing a wavelet transform across the motion trajectories, as described in [8]. The main issue, in this case, is that the motion vectors are not available before acquisition, and an iterative and computationally demanding estimation procedure should be carried out.

While the CS approach can substantially reduce the number of measurements in the acquisition phase, it cannot remove the inefficiencies encountered in the subsequent analysis stage, where the video needs to be fully reconstructed from its random measurements in order for high-level descriptors to be computed. In other words, performing CS and video analysis in a disjoint way results in an overall increase of bit rate, since to obtain a good reconstruction quality of the whole frame (and therefore better features) a higher number of measurements needs to be coded and transmitted with a lower distortion [9]. In this paper, we argue that, for some specific analysis tasks, integrating the three basic modules described above (sensing, coding and analysis) into a “*joint compressive video coding and analysis*” scheme can bring considerable advantages over the case where CS and analysis are treated as disjoint tasks. We illustrate this integration for a particular video analysis setting, namely the tracking of moving objects in a video sequence characterized by a slowly varying background, which is typical of video surveillance scenarios. For this specific application, we show that an analysis-aware scheme devised to perform CS in the light of the subsequent analysis process can achieve better performance than a disjoint sensing and analysis method, for a given target bit rate. Specifically, we mix analysis and CS in two ways. First, we observe that some tracking algorithms actually get rid of the frame background when computing the position and the size of the bounding box enclosing the objects in the scene. Thus we do not reconstruct the whole frame but only the foreground, by subtracting the background directly in the projections domain as suggested in [10]. Second, we use the predicted positions and sizes of the bounding boxes computed by the tracking algorithm as a prior information for the CS decoder to direct the reconstruction process. We show that, using this joint approach, the number of measurements to be acquired by a CS architecture can be substantially reduced with respect to a disjoint CS video coding and analysis.

The rest of this paper is organized as follows. Section II provides a brief overview of some key CS ideas, which are a necessary prerequisite for understanding the rest of the paper. In Section III, we illustrate the disjoint CS-and-analysis scheme. Since in this work we focus on the coding aspect, we will describe in detail the CS-based coding of video frames, from which the foreground can be extracted later. We postpone instead the description of the adopted tracking algorithm to Section IV, where we discuss our proposed joint compressive video coding and analysis approach which integrates the tracking results in the decoding stage. Section V compares the results of the two methods in terms

of the quality of reconstructed foreground, which directly impacts the tracking accuracy. In this way, we are able to evaluate the coding performance independently from the specific tracking algorithm implemented. Finally, Section VI concludes the paper with some hints for future research directions.

II. BACKGROUND ON COMPRESSIVE SENSING

Compressive sensing theory asserts that it is possible to perfectly recover a signal from a limited number of incoherent non-adaptive linear measurements, provided that the signal can be represented by a small number of nonzero coefficients in some basis expansion. Suppose we can write the signal $\mathbf{x} \in \mathbb{R}^N$ to be acquired as $\mathbf{x} = \Phi\boldsymbol{\theta}$, where $\boldsymbol{\theta} \in \mathbb{R}^N$ is a k -sparse vector, i.e. just k out of the N elements of $\boldsymbol{\theta}$ are nonzero. In other words, we are making the assumption that \mathbf{x} can be represented by a few basis vectors in the orthonormal basis Φ using the basis expansion coefficients $\boldsymbol{\theta}$. Let $\mathbf{y} \in \mathbb{R}^n$, $n < N$, denote a number of linear random projections (measurements) obtained as $\mathbf{y} = \mathbf{A}\mathbf{x}$. If the measurement matrix \mathbf{A} satisfies a *Restricted Isometry Property* (RIP) [2], it can be shown [3] that solving the following optimization problem:

$$\text{minimize } \|\boldsymbol{\theta}\|_1 \quad \text{s.t. } \mathbf{y} = \mathbf{A}\Phi\boldsymbol{\theta} \quad (1)$$

is equivalent to finding the sparsest vector $\hat{\boldsymbol{\theta}}$ that fulfils the constraint $\mathbf{y} = \mathbf{A}\Phi\hat{\boldsymbol{\theta}}$, provided that the number of measurements satisfies $n \geq Ck \log(N/k)$, where C is a well-behaved constant. In practice, the RIP holds whenever the columns of matrix \mathbf{A} are incoherent with the basis Φ in which the signal is sparse, and it turns out that sampling the entries of matrix \mathbf{A} from a Gaussian distribution with zero mean and variance $1/N$ provides a measurement basis which is incoherent with overwhelming probability with any other given basis.

In most practical applications, measurements are affected by noise. In this case, we can express the noisy measurements as $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{z}$, where the noise amplitude is assumed to be bounded, i.e. $\|\mathbf{z}\|_2 \leq \sigma$. An approximation of the original signal \mathbf{x} can be obtained by solving the following problem:

$$\text{minimize } \|\boldsymbol{\theta}\|_1 \quad \text{s.t. } \|\mathbf{y} - \mathbf{A}\Phi\boldsymbol{\theta}\|_2 \leq \sigma. \quad (2)$$

A very common situation in which this occurs is when the measurements are quantized. If a uniform quantizer with quantization step Δ is used, then the quantization error for each measurement behaves like a uniformly distributed random variable on the interval $[-\Delta/2, \Delta/2]$, so that we can recover the original signal with very small error by solving Problem (2) by setting the value of σ^2 equal to [11]:

$$\sigma^2 = n \frac{\Delta^2}{12} + 2\sqrt{n} \frac{\Delta^2}{6\sqrt{5}} \quad (3)$$

Problem (2) is an instance of a second order cone program (SOCP) [12] and can be solved in $O(n^3)$ computations. Nevertheless, several fast algorithms have been proposed in the literature that attempt

to find a solution to (2). In this work, we adopt the SPGL1 algorithm [13], which is specifically designed for large scale sparse reconstruction problems.

A recent work by Candes et al. [14] has shown that, by inserting proper weights into the objective function in (2), one can enhance the quality of the reconstruction, reducing at the same time the number of required measurements. The rationale behind this approach is that, by using some a priori knowledge about the support and the values of the sparse signal, it is possible to direct the reconstruction process towards the actual nonzero values. Thus, the problem that has to be solved in this case is:

$$\text{minimize } \|\mathbf{W}\boldsymbol{\theta}\|_1 \quad \text{s.t.} \quad \|\mathbf{y} - \mathbf{A}\boldsymbol{\Phi}\boldsymbol{\theta}\|_2 \leq \sigma, \quad (4)$$

where \mathbf{W} is a diagonal matrix with weights $\mathbf{w} = [w(1) \dots w(N)]$ on the diagonal. In [14] it is shown that the weights should be chosen to be inversely proportional to the expected signal magnitude:

$$w(i) = \frac{1}{|\theta(i)| + \epsilon} \quad (5)$$

where ϵ is added at the denominator to avoid division by zero for exactly sparse signals. Intuitively, the modified objective function favors a solution with non-zero values corresponding to the indices where w_i is small. Of course, it is impossible to know in advance the magnitude of the signal coefficients before actually reconstructing them. For this reason, the authors of (5) propose a *reweighing* procedure to learn iteratively these weights as the signal is being reconstructed. In this work, however, we will set the weights in (5) according to the a priori knowledge of the signal support estimated in the analysis stage.

III. DISJOINT COMPRESSIVE VIDEO CODING AND ANALYSIS

In this section we consider a video analysis scenario in which the analysis (in this case, object tracking) is performed independently after the CS acquisition phase. In other words, random projections of the video frames are first captured by means of a compressive sensing device [5], [4]; then, the video sequence is reconstructed and the foreground is extracted and processed for analysis in a traditional fashion. The quality of the extracted foreground, therefore, will be the same as the whole quality of the frame. Before proceeding, we want to point out that our setting fits particularly well to a scenario where a conventional video acquisition (e.g. CCD arrays) and coding scheme (e.g. H.264/AVC) is unfeasible or too expensive to be applied. This may be the case, for instance, of infrared video-surveillance, where the high cost of acquisition devices has hindered their widespread diffusion in most non-professional commercial systems. Conversely, we observe that if the video signal can be acquired directly in the pixel domain, it does not seem to be reasonable to compute

random projections and discarding the pixel values. In fact, in this case, other conventional analysis methods shall be adopted, different from the one presented in the rest of this paper.

We represent the video to be encoded as a sequence $\{\mathbf{x}_1, \dots, \mathbf{x}_t, \dots, \mathbf{x}_{N_F}\}$ of N_F frames, where the subscript refers to the temporal index. Each frame is represented as a column vector $\mathbf{x}_t \in \mathbb{R}^N$, where N is the number of pixels in a frame. We can encode each frame of the sequence by computing the projections through a suitable measurement matrix \mathbf{A} and applying traditional encoding strategies such as PCM (Pulse-Code Modulation) scalar quantization or predictive coding adopting DPCM (Differential Pulse-Code Modulation) [15]. At the decoder, video frames can be recovered from their quantized random projections by means of ℓ_1 optimization as described in Section II. Video frames are not sparse in the pixel domain. Therefore a suitable spatial transform needs to be used. In this work we adopt a 2D wavelet transform to cope with spatial intra-frame correlations. This kind of transform has proved to be very effective in traditional image and video coding, since it provides a sparse signal representation and computationally efficient algorithms. Moreover, inter-frame correlation can be exploited by means of a temporal transform. In this paper we consider two different transforms: a Haar wavelet transform and a signal-adaptive Karhunen-Love Transform (KLT).

Given the general framework described above, we can envisage three different acquisition and coding strategies that differ based on the adopted measurement matrix:

- *Variable matrix.* The projections of each video frame are computed by means of a different measurement matrix. Each projection refers to samples belonging to an individual frame. As we shall describe later, with this strategy no temporal correlation can be exploited at the encoder side, even if other non-coding aspects may benefit from this strategy (e.g. secrecy [16]).
- *Fixed matrix.* The projections of each video frame are computed by means of the same measurement matrix. Each projection refers to samples belonging to an individual frame. The correlation between the projections of consecutive frames can be exploited in order to build an encoding scheme that allows considerable savings in the number of bits that need to be transmitted to the decoder.
- *Global matrix.* Consecutive frames are buffered to form a group of pictures (GOP) and random projections are computed from this three-dimensional volume of data. Each projection refers to samples belonging to all the frames in the GOP. To the authors' knowledge, there is no practical hardware device capable of directly acquiring such spatio-temporal measurements, without the need to acquire pixel values first. Nevertheless, we decided to include this system in our comparisons.

A. Variable matrix

The first method that we consider consists in adopting a different measurement matrix $\mathbf{A}_t \in \mathbb{R}^{n \times N}$ for each frame. The entries of the matrices \mathbf{A}_t are sampled from a Gaussian distribution $\mathcal{N}(0, 1/N)$ using a random seed that is known at the decoder side. First, a number of linear random projections $\mathbf{x}_t^p \in \mathbb{R}^n$, $n < N$, is computed as $\mathbf{x}_t^p = \mathbf{A}_t \mathbf{x}_t$ (for notational convenience, we will use the superscript p hereafter to denote random projections). Second, these random projections \mathbf{x}_t^p are encoded using a PCM scheme with an optimal uniform scalar quantizer with step size Δ . Finally, the resulting quantized frame projections $\hat{\mathbf{x}}_t^p$ are sent to the decoder. We notice that temporal correlation between consecutive frames cannot be exploited *at the encoder side*. In fact, due to the use of a variable matrix, there is no temporal correlation between the projections of consecutive frames.

Exploiting the results about compressive sensing presented in Section II, we can compute an estimate of the t -th frame \mathbf{x}_t as $\hat{\mathbf{x}}_t = \Phi \hat{\boldsymbol{\theta}}$, where Φ is the 2D wavelet transformation matrix and $\hat{\boldsymbol{\theta}}$ is the solution of the following optimization problem:

$$\text{minimize } \|\boldsymbol{\theta}\|_1 \quad \text{s.t.} \quad \|\hat{\mathbf{x}}_t^p - \mathbf{A}_t \Phi \boldsymbol{\theta}\|_2 \leq \sigma \quad (6)$$

In order to further leverage signal sparsity, hence obtaining an improved reconstruction for the same number of projections, we can exploit the temporal correlation between consecutive frames at the *decoder side*. As suggested in [17], we can define a joint measurement matrix \mathcal{A}_t as

$$\mathcal{A}_t = \begin{bmatrix} \mathbf{A}_{t-G+1} & 0 & \cdots & 0 \\ 0 & \mathbf{A}_{t-G+2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{A}_t \end{bmatrix} \quad (7)$$

and then perform joint reconstruction of a GOP of G consecutive frames using a 3D wavelet transformation matrix Ψ for the frame ensemble. The projected frames are buffered and stored in a vector $\hat{\mathbf{x}}_{t-G+1:t}^p = [(\hat{\mathbf{x}}_{t-G+1}^p)^T \cdots (\hat{\mathbf{x}}_{t-1}^p)^T (\hat{\mathbf{x}}_t^p)^T]^T$. Then the whole GOP is recovered as $\hat{\mathbf{x}}_{t-G+1:t} = \Psi \hat{\boldsymbol{\theta}}$, where $\hat{\boldsymbol{\theta}}$ is the solution of the following optimization problem:

$$\text{minimize } \|\boldsymbol{\theta}\|_1 \quad \text{s.t.} \quad \|\hat{\mathbf{x}}_{t-G+1:t}^p - \mathcal{A}_t \Psi \boldsymbol{\theta}\|_2 \leq \sigma \quad (8)$$

B. Fixed matrix

An alternative option for acquiring and encoding the video sequence consists in adopting the same measurement matrix \mathbf{A} for all the frames. Similarly to the previous case, we can define a fixed joint

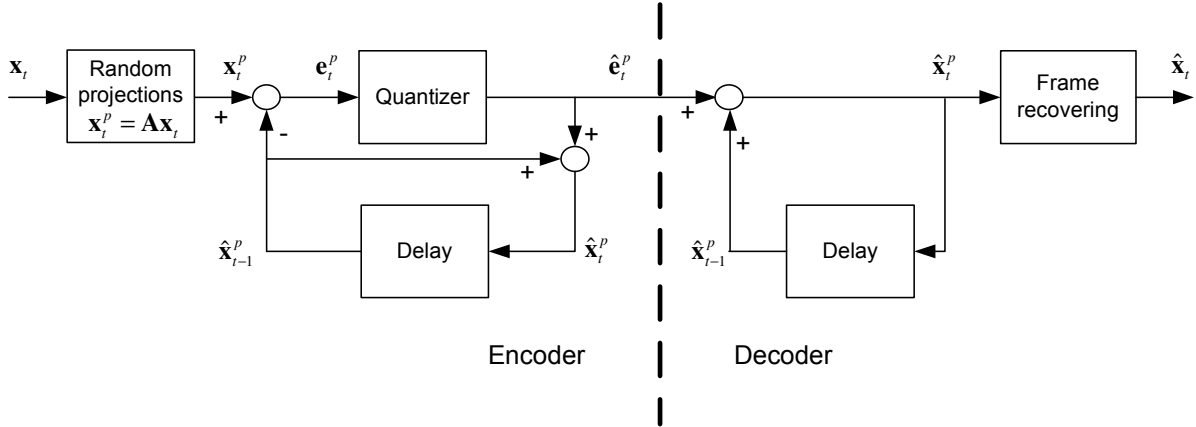


Fig. 1. Block diagram of the fixed matrix DPCM coding scheme.

measurement matrix \mathcal{A} given by:

$$\mathcal{A} = \begin{bmatrix} \mathbf{A} & 0 & \cdots & 0 \\ 0 & \mathbf{A} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{A} \end{bmatrix} \quad (9)$$

With respect to the approach that uses a variable matrix, in this case the joint measurement matrix has a more regular structure, because of the presence of multiple replicas of the matrix \mathbf{A} . Therefore, it is characterized by a weaker incoherence with respect to the 3D wavelet matrix Ψ , thus suggesting that a larger number of measurements is needed to achieve the same reconstruction quality. Conversely, the most significant benefit of the fixed matrix approach is that, since all the frames of the sequence are projected through the same measurement matrix \mathbf{A} , a DPCM coding scheme such as the one shown in Figure 1 can be used instead of PCM. Therefore a significant performance improvement in terms of coding efficiency can be achieved, due to the lower number of bits/measurement adopted in DPCM vs. PCM [15]. Notice that DPCM cannot be used if the variable matrix approach is adopted. In fact, the key point at the base of DPCM is that, due to correlation between \mathbf{x}_{t-1}^p and \mathbf{x}_t^p , the prediction residual \mathbf{e}_t^p has a smaller variance with respect to \mathbf{x}_t^p and, as a consequence, a smaller distortion is obtained by quantizing the prediction residuals rather than the frame projections. Therefore, if a different measurement matrix is used for each frame, \mathbf{x}_{t-1}^p and \mathbf{x}_t^p are uncorrelated, so that no concrete benefit can result from the quantization of the prediction residuals.

Since it seems reasonable to assume the prediction residuals to be sparser than the original frames, one may think of exploiting compressive sensing in order to compute an estimate of the prediction residuals $\hat{\mathbf{e}}_t$ from their quantized projections $\hat{\mathbf{e}}_t^p$ and finally recover the frame as $\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t-1} + \hat{\mathbf{e}}_t$.

Although the prediction residuals are easier to recover thanks to their stronger sparsity, this approach is not characterized by an improved performance. Instead, in this case the frame reconstruction error suffers from drift. The reason of the error drift consists in the fact that the signal recovery block and the DPCM loop at the decoder are swapped with respect to the encoder, violating the fundamental requirement that identical predictions must be computed at both ends. Therefore, in order to exploit DPCM coding, we proceed as follows. First, we compute the quantized frame projections $\hat{\mathbf{x}}_t^p$ starting from the projected prediction residual $\hat{\mathbf{e}}_t^p$. Second, we solve an ℓ_1 norm optimization problem in order to recover the frame.

Moreover, adopting a fixed measurement matrix enables to implement alternative solutions for the frame reconstruction module. In fact, temporal correlation between consecutive frames might be exploited by means of a signal adaptive Karhunen-Love Transform (KLT) [15], instead of a fixed temporal wavelet transform. If properly setup, the KLT might provide a sparser representation of the underlying video signal, so as to achieve improved reconstruction for the same number of measurements. In conventional coding schemes, the main drawback related to the use of the KLT is that it requires the knowledge of the autocorrelation matrix of the signal to be encoded beforehand, and therefore it is signal dependent. Nevertheless it can be shown that in the coding scheme presented in this paper, an approximation of the KLT can be learnt at the decoder side from the available quantized random projections. Let $\mathbf{X}_t \in \mathbb{R}^{N \times G}$ denote the matrix obtained by stacking as columns the vectors $\mathbf{x}_{t-G+1}, \dots, \mathbf{x}_{t-1}, \mathbf{x}_t$ and let \mathbf{X}_t^p be the equivalent of \mathbf{X}_t in the projections domain. Since all the frames are projected through the same measurement matrix \mathbf{A} , we have that $\mathbf{x}_t^p = \mathbf{A}\mathbf{x}_t \forall t$, which can be also written as $\mathbf{X}_t^p = \mathbf{A}\mathbf{X}_t \forall t$. Each row of the matrix \mathbf{X}_t represents the temporal evolution of the intensity values of each pixel and can be thought as a different realization of a discrete time random process. The autocorrelation matrix of this process can be expressed as $R = E[\mathbf{X}_t^T \mathbf{X}_t]$, where the symbol $E[\cdot]$ denotes the expected value. The KLT transform matrix is defined as $[\mathbf{u}_1 | \mathbf{u}_2 | \dots | \mathbf{u}_G]$, where \mathbf{u}_k , $k = 1, 2, \dots, G$ are the eigenvectors of R . Even if the original frames are unknown, we can estimate the autocorrelation matrix starting from the random projections. In fact, neglecting the quantization noise and remembering that the measurement matrix satisfies the RIP so that $\mathbf{A}^T \mathbf{A} \approx \mathbf{I}_N$, we have that

$$E[(\mathbf{X}_t^p)^T \mathbf{X}_t^p] = E[(\mathbf{A}\mathbf{X}_t)^T (\mathbf{A}\mathbf{X}_t)] = E[\mathbf{X}_t^T \mathbf{A}^T \mathbf{A} \mathbf{X}_t] \approx E[\mathbf{X}_t^T \mathbf{X}_t] \quad (10)$$

i.e. the autocorrelation matrix of the underlying video signal can be approximated by the autocorrelation matrix of the available random projections.

C. Global matrix

As a last option, we can perform joint encoding and reconstruction of a group of frames, using a global matrix $\mathcal{A} \in \mathbb{R}^{Gn \times GN}$ whose entries are sampled from a Gaussian distribution $\mathcal{N}(0, 1/(GN))$. G consecutive frames $\mathbf{x}_{t-G+1} \dots \mathbf{x}_{t-1} \mathbf{x}_t$ are buffered in a vector $\mathbf{x}_{t-G+1:t} \in \mathbb{R}^{GN}$, projected through the matrix \mathcal{A} and finally encoded with a PCM scheme. As in the case of variable matrix, the group of frames is reconstructed using a 3D wavelet transformation matrix Ψ , but neither a DPCM coding scheme nor a signal-adaptive KLT can be used in this case. We will show that, even if matrices \mathcal{A} and Ψ are more incoherent than when a fixed matrix is used, the coding gain provided by a DPCM scheme still outperforms the variable and global matrix approaches.

D. Video analysis

Conventionally, video analysis is conceived as a disjoint task with respect to video coding. Therefore, the video frames needs to be reconstructed in the pixel domain and then fed in input to some video analysis algorithm (e.g. object tracking). In this case, the accuracy of the results produced by the analysis is affected by the distortion introduced in the recovered video signal. Despite of the specific measurement matrix adopted, in Section V we show that this approach requires a large bitrate to achieve a distortion level suitable to enable further processing. This is due to the fact that most of the recovered information is discarded by the analysis module. The joint compressive video coding and analysis scheme presented in the following section has been designed to cope with this issue.

IV. JOINT COMPRESSIVE VIDEO CODING AND ANALYSIS

In some applications we do not need to recover the original frames. For example, many automatic video surveillance tasks, e.g. object tracking, can be carried out with the only knowledge of the foreground scene. In this section, we propose to adapt the coding scheme shown in Figure 1 to directly compute the foreground images without recovering the original frames. Foreground images are the result of the subtraction of the local background from the original images. The pixels corresponding to the background region are equal to zero or close to zero. Thus, the foreground images are supposedly sparser than the original frames, and we can expect them to be easier to recover from their random projections, thus reducing the bitrate needed to accomplish the analysis task.

We will refer to the coding architectures presented in this section as joint compressive video coding and analysis schemes. The attribute *joint* refers to two distinct aspects: first, we decode only the minimum amount of data needed to perform video analysis, namely foreground objects; second, at least for some of the schemes presented below, video decoding and object tracking are tightly

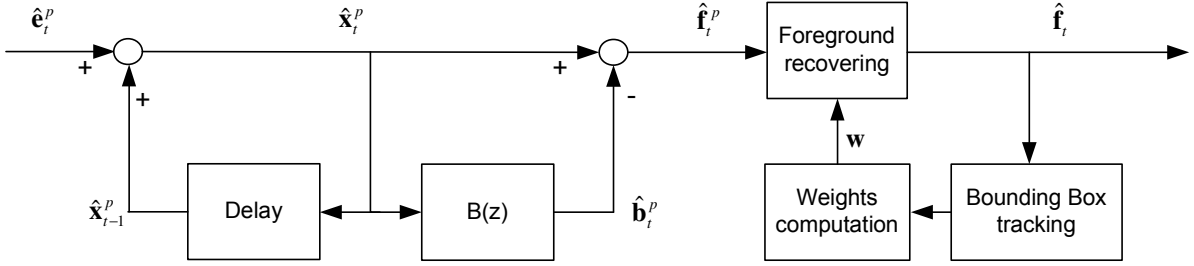


Fig. 2. Block diagram of the proposed joint compressive video coding and analysis scheme.

coupled, in the sense that the result of tracking is used to improve the foreground recovery, which is in turn used for object tracking.

The proposed coding architecture consists in using the fixed matrix DPCM encoder depicted in Figure 1 combined with the decoder shown in Figure 2. As in the previous case, in the first decoding stage the quantized frame projections \hat{x}_t^p are computed starting from the quantized residual projections \hat{e}_t^p . The foreground image is then estimated directly in the projections domain following the procedure detailed later in Section IV-A. Finally, weighted ℓ_1 optimization is exploited in order to recover the foreground image \hat{f}_t from its projections \hat{f}_t^p , as explained in Section IV-B. The prior information required for the computation of the weights is inferred from the previous recovered foreground image \hat{f}_{t-1} . To be more precise, we identify the foreground objects and track the motion of the bounding boxes enclosing them by means of particle filtering. This allows to obtain a sort of motion-compensated estimate of the current foreground image, which can be effectively used as prior information in order to solve the recovery problem through weighted ℓ_1 optimization. Details about the adopted tracking scheme are provided in Section IV-C. The algorithm used for the computation of the weights is instead described in Section IV-D.

A. Background subtraction

Background subtraction [18] is a widely used approach for detecting moving objects in videos from static cameras. Let $\{\mathbf{x}_1, \mathbf{x}_2 \dots \mathbf{x}_t \dots \mathbf{x}_{N_F}\}$ be the acquired frame sequence and let $\hat{\mathbf{b}}$ be an estimate of the scene slowly varying background. Then, at each time instant t , the image of the foreground can be computed as $\hat{f}_t = \mathbf{x}_t - \hat{\mathbf{b}}$. Actually, the background model cannot be fixed but it must adapt to illumination and motion changes. Thus, it must be continuously updated as new frames are acquired. A very simple and computationally efficient method to do this is the running average method [18]:

$$\hat{\mathbf{b}}_t = \alpha \mathbf{x}_{t-1} + (1 - \alpha) \hat{\mathbf{b}}_{t-1} \quad (11)$$

where $\alpha \in [0, 1]$ is a parameter the background adaptation rate. This method can also be directly implemented in the projections domain. Let $\mathbf{x}_t^p = \mathbf{A}\mathbf{x}_t$ be the current frame projections and let $\hat{\mathbf{b}}_t^p = \mathbf{A}\hat{\mathbf{b}}_t$ be the background projections. It comes out that the foreground projections are easily computed as:

$$\hat{\mathbf{f}}_t^p = \mathbf{A}\hat{\mathbf{f}}_t = \mathbf{A}(\mathbf{x}_t - \hat{\mathbf{b}}_t) = \mathbf{A}\mathbf{x}_t - \mathbf{A}\hat{\mathbf{b}}_t = \mathbf{x}_t^p - \hat{\mathbf{b}}_t^p \approx \hat{\mathbf{x}}_t^p - \hat{\mathbf{b}}_t^p \quad (12)$$

while the background projections can still be updated with the running average method [10], without the need of recovering the pixel domain representation of the background itself:

$$\hat{\mathbf{b}}_t^p = \alpha\hat{\mathbf{x}}_{t-1}^p + (1 - \alpha)\hat{\mathbf{b}}_{t-1}^p. \quad (13)$$

In Figure 2 background prediction is denoted by the recursive filter $B(z)$, whose transfer function is

$$B(z) = \frac{\alpha z^{-1}}{1 - (1 - \alpha)z^{-1}}. \quad (14)$$

We point out that, despite of the particular choice of background subtraction algorithm made in this paper, any other background subtraction technique can be use, provided it is linear.

B. Foreground recovering

The objective of the foreground recovery module is to reconstruct the foreground image \mathbf{f}_t starting from its quantized random projections $\hat{\mathbf{f}}_t^p$. This can be done exploiting temporal correlation by means of a linear transformation. Given a joint measurement matrix \mathcal{A}_t and a 3D wavelet transformation matrix Ψ , we can solve the following optimization problem, analogous to Problem (8):

$$\text{minimize } \|\boldsymbol{\theta}\|_1 \quad \text{s.t.} \quad \|\hat{\mathbf{f}}_{t-G+1:t}^p - \mathcal{A}_t \Psi \boldsymbol{\theta}\|_2 \leq \sigma \quad (15)$$

where $\hat{\mathbf{f}}_{t-G+1:t}^p = [(\hat{\mathbf{f}}_{t-G+1}^p)^T \dots (\hat{\mathbf{f}}_{t-1}^p)^T \quad (\hat{\mathbf{f}}_t^p)^T]^T$ is a vector obtained by stacking G column vectors representing the projections of the foreground images.

Leveraging the results about weighted ℓ_1 optimization presented in Section II, we propose an alternative approach to exploit temporal correlation. Rather than striving for a sparser representation of the signal, we attempt to enhance the reconstruction performance inferring information about the current foreground image from the previous one and using such information to compute the weights that might help solving the recovery problem. To be more precise, an estimate of the foreground image is computed as $\hat{\mathbf{f}}_t = \Phi \hat{\boldsymbol{\theta}}$, being $\hat{\boldsymbol{\theta}}$ the solution of the following optimization problem:

$$\text{minimize } \|\mathbf{W}\boldsymbol{\theta}\|_1 \quad \text{s.t.} \quad \|\hat{\mathbf{f}}_t^p - \mathbf{A}\Phi\boldsymbol{\theta}\|_2 \leq \sigma \quad (16)$$

where \mathbf{W} is a diagonal matrix with the weights $\mathbf{w} = [w(1) \dots w(N)]$ on the diagonal, Φ is an orthonormal 2D wavelet transformation matrix and $\boldsymbol{\theta} = \Phi^T \mathbf{f}_t$ is the vector representing \mathbf{f}_t in the wavelet domain. In the following we explain how the weight vector \mathbf{w} is computed, by tracking the bounding box of the moving objects.

C. Bounding box tracking

In order to exploit weighted ℓ_1 optimization to solve the recovery problem we need a way for inferring prior information about the current foreground image from the previously decoded frames. Our solution consists in identifying the foreground objects and tracking the motion of the bounding boxes enclosing them, so that we can estimate the position of the objects in the current frame exploiting the past ones.

Let us assume that, at some time instant, the foreground image has been correctly obtained. In order to initialize the tracking algorithm, the absolute value of the foreground image is thresholded and the connected foreground pixels are labeled as being part of the same blob, so that the identified blobs represent the objects in the scene. In order to make the system robust to over-segmentation, we implemented a simple algorithm that merges blobs on the basis of a *Virtual Merge Evaluation* criteria [19]. The next step consists in assigning each blob to one of the detected objects. In other words, we need to link each of the blobs identified in the current frame with the blob representing the same moving object in the previous frame. In this way we can successfully track the motion of the object across frames. The match is performed by associating each blob with the object that best describes it in terms of position and size.

Object tracking is performed by means of particle filtering. The problem can be formulated as the estimation of the a posteriori probability distribution of a random variable \mathbf{z}_t , representing the state of the system at time t (e.g. object position and velocity), given the available observations $\{\mathbf{y}_1 \dots \mathbf{y}_t\}$ (e.g. available video data). Particle filters model the a posteriori distribution of the state as a finite set of particles, each associated to a state vector \mathbf{z}_t^j and a particle weight ω_t^j , which is proportional to the likelihood of the state vector \mathbf{z}_t^j with respect to the current observations. In this work we use a Sequential Importance Resampling (SIR) particle filter implementation [20], which requires the definition of a transition model $P(\mathbf{z}_t|\mathbf{z}_{t-1})$, to define the dynamic evolution of the particle states, and a likelihood function $P(\mathbf{y}_t|\mathbf{z}_t)$, to compute the particle weights.

In the specific application scenario addressed in this work, $\mathbf{z}_t = [\mathbf{c}_t, \mathbf{s}_t, \mathbf{u}_t]^T$ represents the state vector, where the components are the bounding box centroid position ($\mathbf{c}_t \in \mathbb{R}^2$), size ($\mathbf{s}_t \in \mathbb{R}^2$) and velocity ($\mathbf{u}_t \in \mathbb{R}^2$) of the objects to be tracked, while the observation vector \mathbf{y}_t is set equal to the blob representing the object in the recovered foreground $\hat{\mathbf{f}}_t$.

The state transition model expresses the *a priori* knowledge about the motion evolution of the target and provides a prediction based on the past state values. At each time instant, the state of each particle is updated according to the following model, proposed in [21] and constructed upon the

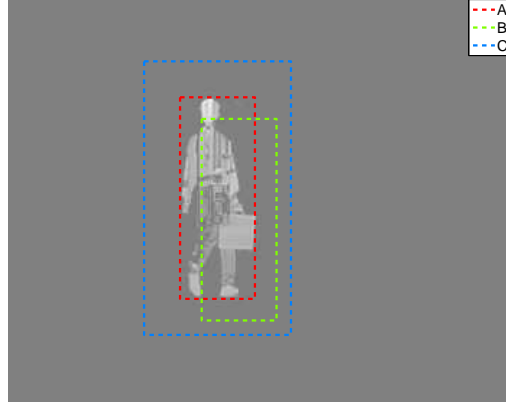


Fig. 3. A pictorial example illustrating the computation of the weights of three particles (A, B and C) from the reconstructed foreground image.

equation of motion:

$$\begin{cases} \hat{\mathbf{c}}_t^j &= \mathbf{c}_{t-1}^j + \mathbf{u}_{t-1}^j \Delta_T + \xi_c \\ \hat{\mathbf{s}}_t^j &= \mathbf{s}_{t-1}^j + \xi_s \\ \hat{\mathbf{u}}_t^j &= \mathbf{u}_{t-1}^j + \xi_u \end{cases} \quad (17)$$

where $[\hat{\mathbf{c}}_t^j, \hat{\mathbf{s}}_t^j, \hat{\mathbf{u}}_t^j]^T$ is the predicted state vector associated with the j -th particle, Δ_T is the time sample interval and ξ_c, ξ_s, ξ_u are random terms which provide the system with a diversity of hypotheses. At each time instant, the bounding box centroid position, size and velocity are estimated according to the following equations:

$$\begin{cases} \mathbf{c}_t &= (\mathbf{c}_{t-1} + \mathbf{u}_{t-1} \Delta_T)(1 - \alpha_c) + \alpha_c \sum_j \omega_t^j \hat{\mathbf{c}}_t^j \\ \mathbf{s}_t &= \sum \omega_t^j \hat{\mathbf{s}}_t^j \\ \mathbf{u}_t &= \mathbf{u}_{t-1}(1 - \alpha_u) + \alpha_u (\mathbf{c}_t - \mathbf{c}_{t-1}) \end{cases} \quad (18)$$

where α_c, α_u are two parameters that adjust the adaptation rates.

The particle weight ω_t^j is computed to be proportional to the matching between the bounding box represented by the state vector \mathbf{z}_t^j and the recovered foreground $\hat{\mathbf{f}}_t$. To be more precise, we use the following formula:

$$\omega_t^j = E_R \cdot d_{BB} \quad (19)$$

Figure 3 gives a pictorial representation of the contribution of each of the two terms composing the likelihood function. The first term represents the energy ratio E_R , which is defined as the ratio between the energy of the portion of the blob contained in the bounding box and the total energy of the blob. This term has the function of penalizing a bounding box if it is not correctly positioned over the blob (B). The second term is the bounding box density d_{BB} , defined as the percentage of pixels contained in the bounding box having an intensity value greater than a fixed threshold. The purpose

of this term is to penalize excessively large bounding boxes (C). In fact, a correctly positioned over-dimensioned bounding box would exhibit maximum energy ratio, as it would completely enclose the blob, but also low density, since many pixels belonging to the bounding box but not to the blob would have negligible intensity values. Therefore, the bounding box would be correctly assigned with a low weight, since it does not provide a satisfactory representation of the blob. On the other hand, a large weight is assigned to a bounding box which correctly matches both the position and size of the blob (A), since it is characterized by maximum energy ratio and very high density.

D. Weights computation based on bounding box

In Section II we have shown that the knowledge of some prior information about the signal \mathbf{f}_t to be recovered enables to improve the quality of the reconstructed signal for the same number of random projections. Such prior information is introduced by means of a vector of weights $\mathbf{w} = [w(1) \dots w(N)]$, where $w(i)$ denotes the weighting factor associated with the i -th coefficient of the vector $\boldsymbol{\theta} = \boldsymbol{\Phi}^T \mathbf{f}_t$ in the 2D wavelet domain. As already mentioned in Section II, $w(i)$ should be (ideally) made proportional to the inverse of the absolute value of the coefficient θ_i [14]. Since θ_i is not available, we propose to set the weights needed for the recovering of the foreground image at time t based on the estimated bounding boxes of the previous frame.

To be more precise, a window capturing the likely foreground position in the current frame is derived from the predicted bounding box positions and sizes. A window \mathbf{p} is constructed on the basis of the pixel domain representation of the foreground image \mathbf{f}_t and next transformed to another window π defined in the wavelet domain. The i -th window coefficient $p(i)$ is set according to the position and size of the object bounding box. For pixel locations within the bounding box, we set $p(i) = 1$. For pixel locations outside the bounding box, the corresponding $p(i)$ smoothly decays to zero as they get far from the bounding box. We compute the distances $d_x(i)$ and $d_y(i)$ to the nearest vertical and horizontal bounding box border, respectively. Then the window coefficient associated to this pixel is given by $p(i) = e^{-\alpha_W(d_x(i)+d_y(i))}$, where $\alpha_W > 0$ is a parameter related to the rate of decay of the window. We set the value of α_W adaptively based on the reliability of the bounding box prediction provided by the particle filter. To this end, we measure the variance of the particles and adapt the value of α_W accordingly. A smaller variance implies a higher confidence in the estimated bounding box, thus the value of α_W can be increased to achieve a sharper decay. If the bounding box prediction is correct, the window coefficients set equal to one correspond to the pixel locations in the foreground image that contain the foreground object.

If more than one object is present in the scene, we compute a different weighting window from each of the bounding boxes. A global weighting window is then obtained as the element-wise sum

of the individual windows.

The window \mathbf{p} resulting from this procedure is related to the pixel domain representation of the signal to be recovered. Since we exploit sparsity of the signal in the wavelet domain, we need the window coefficients to be related to the wavelet domain representation of the foreground image θ . The wavelet domain window π can be computed just by applying a suitable transformation to the weighting window \mathbf{p} . For example, Figures 4(a) and 4(b) show how the weighting window transformation is performed when a 2D wavelet transform with two decomposition levels is adopted. Figure 4(a) depicts the window coefficients in the pixel domain for a given location of the estimated bounding box. The window is replicated in the seven low-resolution versions of the image in the wavelet domain, as shown in Figure 4(b). This kind of approach, even if quite simple, allows to directly compute a representation of the window π in the wavelet domain and it can be applied independently from the actual window shape.

The weights necessary to solve the weighted ℓ_1 optimization problem are finally computed by taking the inverse of each coefficient of the window, so that foreground pixel locations which are likely to contain the foreground objects are associated with low weights. Precisely, the i -th weight $w(i)$ is computed as:

$$w(i) = \frac{1}{\pi(i) + \epsilon} \quad (20)$$

where the parameter $\epsilon > 0$ has been introduced in order to provide stability.

E. Weight computation based on object silhouette

Even if the bounding box position is estimated correctly, the weighting window described in Section IV-D can be too conservative in many cases. In fact, it is very unlikely that the foreground object image overfills the bounding box enclosing it and, depending on the object shape, it may happen that the occupied area is very small. For this reason, we investigated an alternative method for the weighting window generation. The underlying idea is to extract from the recovered foreground image of the previous frame the silhouettes of the objects and translate their locations according to the predicted motion of the corresponding bounding box. The pixel domain representation of the window is obtained as the result of a sequence of three steps:

- *Silhouette thresholding.* Object silhouettes are extracted from the recovered foreground image $\hat{\mathbf{f}}_{t-1}$ by means of a thresholding operation. We set the threshold value equal to $\tau_S = \frac{1}{10} \max\{|\hat{\mathbf{f}}_{t-1}|\}$ and retain the values of the pixels above τ_S . This enables to discard non-zero pixel values of the foreground image introduced by quantization noise.
- *Silhouette translation.* The resulting silhouettes cannot be directly used as a prediction of the current foreground image, because they identify the position of the objects in the previous frame.

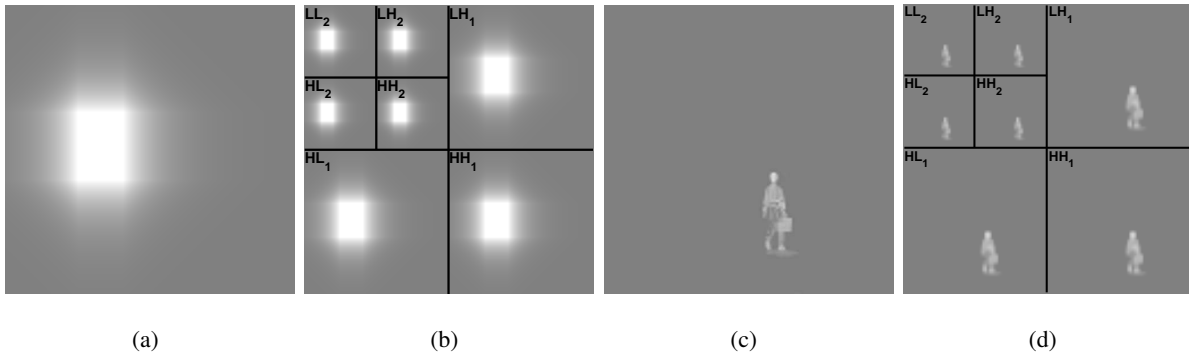


Fig. 4. Example of window transformation: **a.** Bounding box based window in the pixel domain. **b.** Bounding box based window in the wavelet domain. **c.** Silhouette based window in the pixel domain. **d.** Silhouette based window in the wavelet domain.

In order to compensate for the objects motion we compute a one-step ahead prediction of the bounding boxes using the model described in Equation (17) and translate the silhouettes by the vectors associated with the predicted bounding boxes centroid displacements.

- *Silhouette dilation.* In order to obtain robustness with respect to a possibly misplaced bounding box, we apply grayscale dilation to the translated image using a square structuring element of side D [22]. As before, we estimate the reliability of the bounding boxes prediction measuring the variance of the particles and adaptively choose the value of D depending on this measure, increasing D when the variance increases.

An example of a window obtained with this method is shown in Figure 4(c). Eventually, in order to cope with the wavelet domain representation of the foreground image, the window needs to be transformed just as in the bounding box case. This is done by generating opportunely scaled replicas of the weighting window, as shown in Figure 4(d).

V. EXPERIMENTAL RESULTS

We tested the proposed system on two different grayscale video sequences, *hall monitor* and *station*¹, both consisting of 300 frames at CIF resolution and, respectively, 30 and 25 frames per second (fps). Since the analysis task that we consider is ultimately concerned with tracking the locations of foreground objects, we work with video sequences decimated by a factor of 8, so that each 8×8 block is mapped to a single pixel. This allows to reduce the required number of measurements to be acquired and, at the same time, to use off-the-shelf reconstruction algorithms [13] achieving reasonable decoding times. The GOP size G is set to 4 pictures. We fix a quantization step $\Delta = 3.5$

¹The sequences are available at <http://vqm.como.polimi.it/sequences>

to attain an average quantization SNR (Signal to Noise Ratio) of the measurements

$$\text{SNR} = 10 \log_{10} \frac{\|\mathbf{x}_{1:N_F}^p\|_2^2}{\|\mathbf{x}_{1:N_F}^p - \hat{\mathbf{x}}_{1:N_F}^p\|_2^2}, \quad (21)$$

approximately equal to 40dB, where $\mathbf{x}_{1:N_F}^p$ and $\hat{\mathbf{x}}_{1:N_F}^p$ denote, respectively, the random projections of the original and the reconstructed sequence.

We start presenting the results relative to disjoint compressive video coding and analysis, showing that a large bit rate is required in order to achieve a distortion level suitable to enable further processing. Next, we illustrate the results relative to joint compressive video coding and analysis and remark the considerable performance improvement allowed by the proposed approach.

A. Disjoint compressive video coding and analysis

We simulate the acquisition process by means of the fixed, variable and global matrices, as described in Section III, and we vary the fraction $\delta = n/N$ of random projections with respect to the original number of pixels. Figure 5 shows the rate-distortion curves obtained for *hall monitor* and *station* when compressive sensing decoding is disjoint from further analysis, i.e. when we aim at reconstructing the original frames first. On the horizontal axis we indicate the total bit rate R , measured in kbps, which is obtained as $R = \delta \cdot N \cdot B \cdot F$, where B is the number of bits/measurement, and F is the frame rate. The value of B depends on the selected quantization step Δ , on the specific coding scheme and, when DPCM is used, on the amount of inter-frame correlation. In our experiments, the average value of B for *hall monitor* (*station*) is equal to 1.4 (2.2) bits/measurement for DPCM and 7.4 (7.1) bits/measurement for PCM. The markers along the curves correspond to different values of $\delta = 0.1, 0.2, \dots, 0.9$. On the vertical axis we measure the PSNR (Peak Signal to Noise Ratio) of the reconstructed frames, defined as:

$$\text{PSNR}_{FR} = 10 \log_{10} \frac{255^2}{\|\mathbf{x}_{1:N_F} - \hat{\mathbf{x}}_{1:N_F}\|_2^2} \quad (22)$$

where $\mathbf{x}_{1:N_F}$ and $\hat{\mathbf{x}}_{1:N_F}$ denote, respectively, the original and the reconstructed sequence. Note that equation (22) represents the conventional PSNR metrics, as it is computed based on the whole frame area. However, we can reasonably argue that the average distortion in the foreground is the same as the average distortion over the entire frame, since the reconstruction error is uniformly spread over the recovered signal samples. We remark that a good quality of the extracted foreground, related to a high value of PSNR, is fundamental to achieve good results in the subsequent analysis stage [18]. The rate-distortion curves in Figure 5(a) are associated to the different coding schemes discussed in Section III: the fixed matrix approach (Fixed FR-W and Fixed FR-K), which adopts DPCM coding [the last letter in the acronym indicates the adopted temporal transform: Haar wavelet (W) or KLT

(K)]; the variable matrix approach (Variable FR-W) and the global matrix approach (Global FR-W), both adopting PCM coding.

We notice that, in all cases, the entries of the measurement matrix consist of a specific realization of a i.i.d. Gaussian random process. Therefore, the results in Figure 5(a) are obtained by averaging ten rate-distortion curves obtained using different realizations of the measurement matrices. In order to avoid cluttering the figure, we decided not to explicitly show confidence intervals. Nevertheless, the confidence intervals are always very small, typically smaller than ± 0.2 dB. Thus, we can conclude that the results obtained are independent from the specific choice of the adopted measurement matrix.

In addition, we also show in Figure 5(a) the rate-distortion curve obtained using a standard H.264/AVC coding scheme. In order to enable a fair comparison, we configured the H.264/AVC encoder as to achieve low-complexity, targeting a video surveillance scenario. The video sequence is encoded using the baseline profile, a IPPP group of picture, and disabling motion estimation for P-slices (i.e. using motion-compensation with all motion vectors set to zero).

The rate-distortion curves depicted in Figure 5(a) suggest the following considerations:

- We notice that the global matrix approach and the variable matrix approach achieve similar coding efficiency. Despite the block diagonal structure of the latter, the joint measurement matrix defined in (7) is characterized by sufficient incoherence with the 3D wavelet matrix Ψ , so that the RIP is satisfied (see Section II). Therefore, we will only consider the variable matrix approach in the following discussion.
- For the video contents employed in our simulations, the choice of the temporal transform used by the fixed matrix approach does not seem to be crucial. In fact, both the Haar wavelet and the KLT achieve similar results in terms of coding efficiency.
- The fixed matrix approach achieves the same reconstruction quality of the variable matrix approach at a much lower rate. For example, for the *hall monitor* sequence, it is possible to achieve a 60% bitrate reduction at a target PSNR equal to 30 dB. Interestingly, we can observe that for the same number of measurements, i.e. for a given value of δ , the variable matrix approach achieves a better reconstruction. At $\delta = 0.3$, the fixed matrix approach achieves 16 dB while the variable matrix approach 28 dB for both tested sequences. This is justified by the loss of incoherence between the joint measurement matrix \mathcal{A} and the 3D wavelet matrix Ψ when the same fixed matrix \mathbf{A} is used to compose the block-diagonal matrix \mathcal{A} . Of course, the loss in terms of reconstruction quality is more than compensated by the bitrate reduction achievable for the fixed matrix approach thanks to the use of a DPCM scheme to tackle temporal redundancy.
- When compared with H.264/AVC, all CS based schemes exhibit a significant coding efficiency loss. Of course, this comparison needs to be interpreted carefully, since they adopt different data

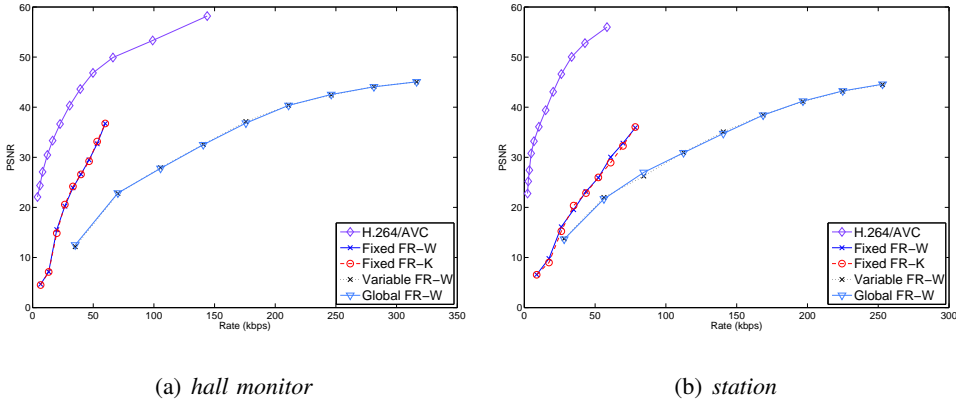


Fig. 5. Rate-distortion curves obtained by the disjoint compressive video coding and analysis schemes. PSNR values refer to the quality of the reconstructed frame.

acquisition methods. In fact, as for H.264/AVC, the video sequence is acquired directly in the pixel domain, while in this paper we assume that the scene is sensed through a compressive sensing camera, ruling out the applicability of conventional video coding schemes. The coding efficiency loss can be attributed to the following factors: first, compressive sensing recovery of sparse signals typically requires a number of measurements larger than the number of non-zero samples, as explained in Section II (similar arguments also hold for compressible signals); second, H.264/AVC tackles spatial redundancy in two ways: block-based transform coding and entropy coding (e.g. CABAC [23]).

Conversely, in the proposed scheme, a suitable spatial transform is applied at the decoder (e.g. a full-frame wavelet transform), but entropy coding is of little help, since the random projections at a given temporal instant are i.i.d. Therefore, we can conclude that with the current understanding of compressive sensing, there is no reason to adopt these kinds of coding schemes when the video sequence can be acquired directly in the pixel domain.

Later in this section we show the results obtained by performing foreground extraction and tracking after the video sequence is reconstructed in the pixel domain.

B. Joint compressive video coding and analysis

In scenarios where the focus is on video analysis, a joint strategy can attain a significant performance improvement. We consider the specific task of object tracking, which is here pursued by first extracting a representation of the foreground objects from the received quantized random projections. We evaluate the coding schemes described in Section IV, and we measure the coding efficiency of the different systems in terms of the quality of the reconstructed foreground as a function of bitrate.

We adopt the same quantization step size Δ and values of δ as in Section V-A. For each value of δ we compute the PSNR of the reconstructed foreground, defined as:

$$PSNR_{FG} = 10 \log_{10} \frac{255^2}{\|\mathbf{f}_{1:N_F} - \hat{\mathbf{f}}_{1:N_F}\|_2^2}, \quad (23)$$

where the actual foreground images $\mathbf{f}_{1:N_F}$ have been obtained by applying the background subtraction method described in Section IV-A to the original sequence $\mathbf{x}_{1:N_F}$.

In order to evaluate the results, we compare the rate-distortion curves obtained using the following methods:

- *FG-N*: We assume that no prior information about the actual foreground image is available, so that the weighted ℓ_1 optimization approach cannot be applied. However, we can still exploit the fact that the foreground image is piecewise smooth in the spatial domain and therefore sparse in a wavelet basis. Hence, given the 2D wavelet transformation matrix Φ , the foreground can be recovered by solving the following optimization problem:

$$\text{minimize } \|\boldsymbol{\theta}\|_1 \quad \text{s.t.} \quad \|\hat{\mathbf{f}}_t^p - \mathbf{A}\Phi\boldsymbol{\theta}\|_2 \leq \sigma \quad (24)$$

- *FG-B*: The foreground image is reconstructed adopting weighted ℓ_1 optimization as formulated in (16). Weights are based on the estimated bounding box obtained exploiting inter-frame dependencies as explained in Section IV-D.
- *FG-S*: The foreground image is reconstructed adopting weighted ℓ_1 optimization as formulated in (16). Weights are obtained based on the object silhouettes as explained in Section IV-E.
- *FG-O*: The foreground image is reconstructed adopting weighted ℓ_1 optimization. Weights are obtained by an oracle, who knows the actual coefficients to be decoded beforehand. Thus, an ideal set of weights can be computed as

$$w(i) = \frac{1}{|\theta(i)| + \epsilon} \quad (25)$$

where $\theta(i)$ is the i -th element of the foreground vector $\boldsymbol{\theta}$ to be reconstructed.

- *FG-W*: The foreground image is reconstructed exploiting temporal correlation that exists between consecutive frames *without* recurring to object tracking. We seek for the solution of problem (15) where the temporal transform matrix is equal to the Haar wavelet.
- *FG-K*: The foreground image is reconstructed as in the previous case, but we solve problem (15) with the temporal transform matrix equal to the KLT estimated as in Section III-B.

We remark the fact that all methods use a 2D wavelet transform to address spatial redundancy. Notice that for the proposed FG-B and FG-S methods, video decoding and analysis are tightly coupled together, since they are based on tracking, respectively, object bounding boxes or silhouettes. We

included the FG-O method in our comparisons as it provides an upper bound for the coding efficiency that can be achieved with the proposed methods, since the weights are optimally selected querying an oracle.

Figure 5 shows the PSNR of the reconstructed foreground obtained by applying the methods listed above, adopting the same quantization step Δ and range of δ values as in Section V-A and averaging over ten different realizations of the fixed measurement matrix. We observe that, as expected, exploiting temporal redundancy is beneficial, due to the residual inter-frame correlation that exists in the sequence of foreground images. By performing decoding and analysis jointly, as in methods FG-B and FG-S, we are able to achieve a higher coding efficiency than simply exploiting a temporal transform, as in FG-W and FG-K (note that, as before, there is no remarkable difference between the FG-W and the FG-K approach). At low bitrates, the PSNR gain can be as large as 7dB. Computing the weights based on the object silhouettes produces consistent gains with respect to the case of using only object bounding boxes, especially at low bitrates, confirming that the system performance is improved if the conservative weighting window described in Section IV-D is substituted with the one proposed in Section IV-E. Nevertheless, there is still a gap between the FG-S and FG-O methods, indicating that the algorithm used for the computation of the weights can be potentially improved. Alternative solutions could hence be investigated in order to improve the system performance and get closer to the solution provided by an oracle.

We emphasize that, in the proposed scheme, weighting is applied only once. This is different from the approach in [14], where CS reconstruction is re-iterated multiple times and, at each step, a refined set of weights is obtained from the partially recovered signal. Our experiments have shown that such a modification does not lead to noticeable gains with respect to the results depicted in Figure 5. This is reasonable, since the initial estimate of the weights provided by the silhouette approach is already satisfactory at the first step, and further refinement steps do not help improving the reconstruction.

C. Comparison of the disjoint and joint video coding and analysis approaches

Figure 7 illustrates, in a single graph, the rate-distortion curves of the coding schemes that attain the best coding efficiency for the disjoint vs. joint video coding and analysis scenarios. As for the disjoint case, we depict the curve corresponding to the fixed matrix method, which employs a 3D wavelet transform to address spatio-temporal redundancy (FR-W). Instead, for the join case, we depict the curve of the FG-S scheme. Note that the former expresses the PSNR of the reconstructed frame, while the latter the PSNR of the reconstructed foreground. Although a direct comparison cannot be made, since the two schemes decode two different signals, we note that at all bitrates, the quality of the reconstructed foreground is remarkably higher. Since the performance of any video analysis task,

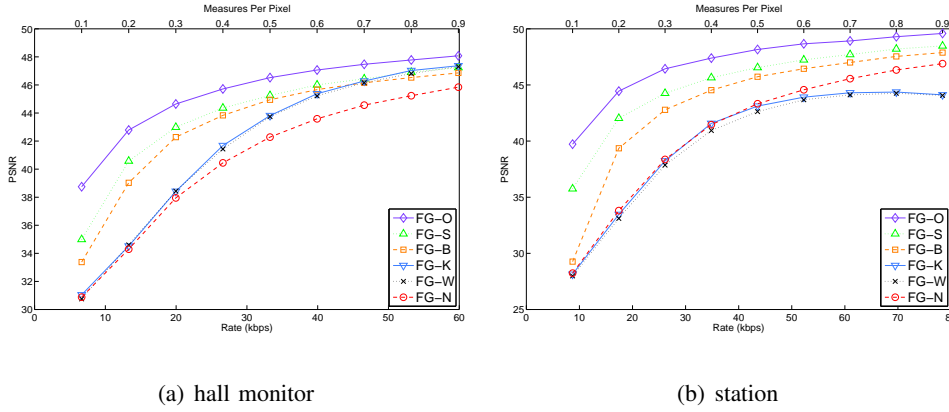


Fig. 6. Rate-distortion curves obtained by the joint compressive video coding and analysis schemes. PSNR values refer to the quality of the foreground images.

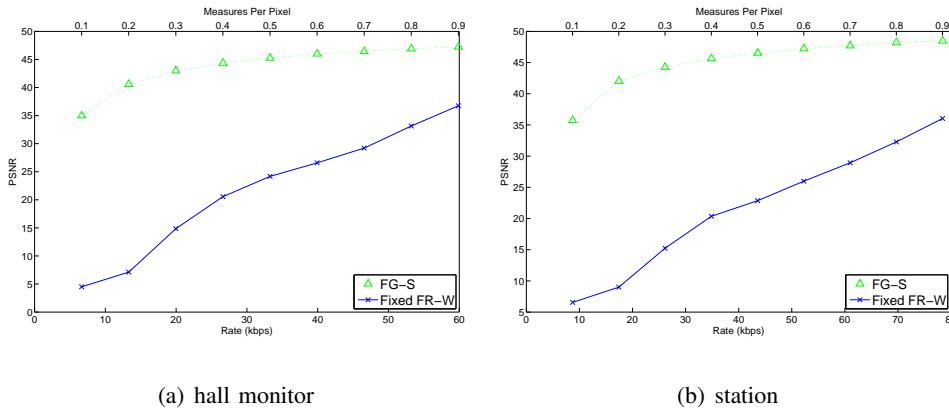


Fig. 7. Rate-distortion curves obtained by the disjoint and joint compressive video coding and analysis schemes.

e.g. object tracking, depends on the quality of the input data, we can argue that a solution that directly reconstructs the foreground images only, instead of the entire video frame, might be preferable.

In order to further support this claim, we evaluated the system performance adopting a different metrics, which is more closely related to the quality of object tracking. Hence, we measure the capability of the various schemes when they are asked to classify each pixel of the video sequence as belonging to the foreground or the background. As ground truth data, we consider the foreground images \mathbf{f} that can be obtained by running the same background subtraction algorithm on the original lossless video sequence. The elements of \mathbf{f} whose magnitude is above a threshold τ are labeled as *positive* (e.g. belonging to the foreground), while the elements whose magnitude is below the threshold are labeled as *negative* (e.g. belonging to the background). Similarly, we threshold the reconstructed foreground $\hat{\mathbf{f}}$ and classify its elements as positive or negative. Given $f(i)$, the i -th element of \mathbf{f} , and its estimate $\hat{f}(i)$, there are four possible outcomes. If $f(i)$ is labeled as positive and it is classified as

positive, it is counted as a *true positive* (e.g. a correctly detected foreground pixel); if it is classified as negative, it is counted as a *false negative* (e.g. a missed foreground pixel). Instead, if $f(i)$ is negative and it is classified as negative, it is counted as a *true negative* (e.g. a correctly detected background pixel); if it is classified as positive, it is counted as a *false positive* (e.g. the pixel is assigned to the foreground when it belongs to the background). We define the probability of detection P_D as the number of true positive elements divided by the total number of positive elements in the ground truth data. Similarly, we define the false positive rate P_{FP} as the number of false positive elements divided by the total number of negative elements. We plot the receiver operating characteristics (ROC) of the classifier, in which P_D is plotted on the vertical axis and P_{FP} is plotted on the horizontal axis. Each value of the threshold τ produces a different point in the ROC graph, so that we can make τ vary in order to build a ROC curve. When comparing two ROC curves produced by different schemes, we prefer the scheme whose ROC is closer to the top-left corner, since it achieves a higher probability of detection P_D at a lower false positive rate P_{FP} .

Figures 8(a) and 8(b) show the ROC curves for the *hall monitor* and *station* sequences. The results have been obtained by running the different coding schemes fixing the number of projections ($\delta = 0.2$). Figures 8(a) and 8(b) compare the performance of systems that reconstruct the foreground images directly (joint compressive video coding and analysis), with a system that performs video decoding followed by background subtraction in the pixel domain (disjoint compressive video coding and analysis). A zoomed view of the top-left corner of Figures 8(a) and 8(b) is illustrated in Figures 8(c) and 8(d) for ease of comparison.

We notice the same behavior already observed when comparing rate-distortion curves. If the foreground is extracted after the video sequence is decoded, as in FR-W, the corresponding ROC curve demonstrates a very poor performance. Conversely, all the methods that reconstruct the foreground directly achieve much better results. In addition, those schemes that couple object tracking with foreground decoding, e.g. FG-B and FG-S, outperform other schemes that exploit temporal redundancy in the foreground images without performing object tracking, e.g. FG-W and FG-K. We also observe that the performance of the FG-S scheme is very close to that of FG-O.

VI. CONCLUSIONS

This paper investigates the potential coding gain that can spring out from jointly designing the processes of video acquisition, compression and analysis. The main tool adopted is the compressive sensing theory, whose principles and applications have been largely described in recent literature. The core tenet of CS asserts that it is possible to acquire certain classes of signals with a substantially smaller number of measurements with respect to conventional sensing devices, by blending together

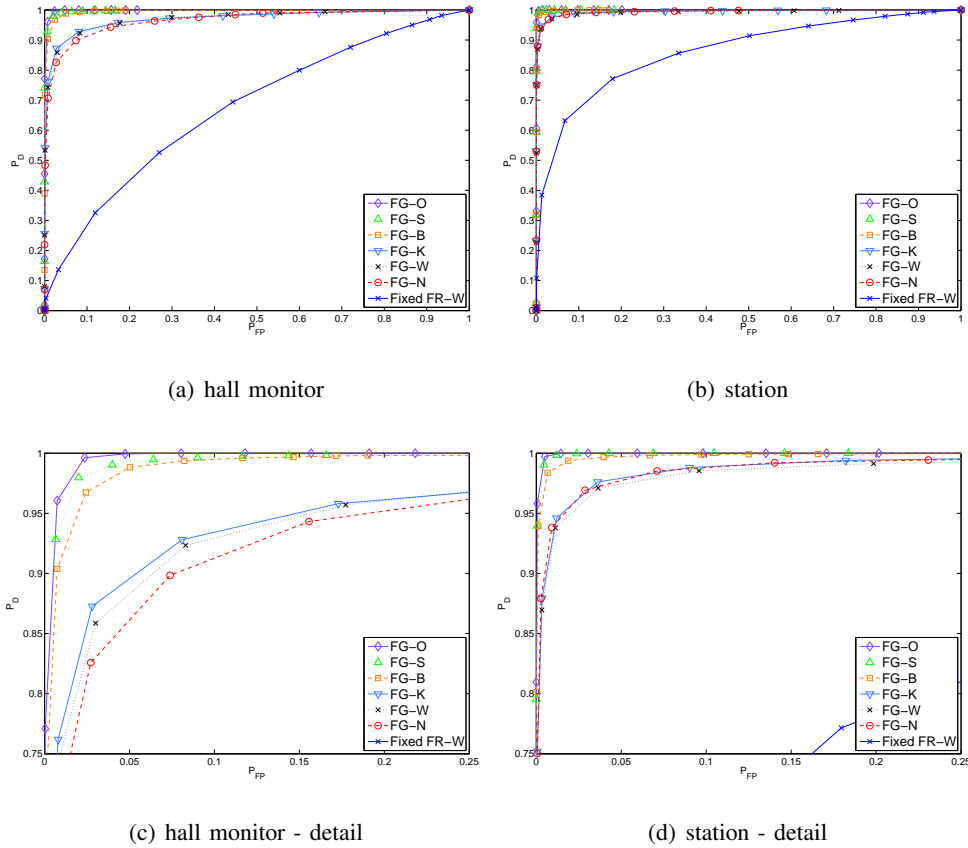


Fig. 8. ROC curves illustrating the detection of the tracked foreground for $\delta = 0.2$.

acquisition and compression. However, there is not any such clear comprehension of how to embed *signal analysis* tasks in this scheme and, in particular, of the potential impact of a joint design in terms of coding efficiency. Our main contribution is to show the advantages of a compressive video coding and analysis scheme over a disjoint approach for a specific analysis scenario, where object tracking is pursued by means of a CS acquisition device. We embed analysis and compressive sensing in two ways. First, we give up reconstructing the whole frames, as what it is really needed in the analysis is the foreground only; second, we feed the information produced in the tracking stage back to the decoding module, where it is used as prior information to direct the reconstruction process. In this way, we achieve a considerable bit rate reduction with respect to the disjoint scheme.

These results suggest that an analysis-aware design of acquisition, coding and signal recovery can produce significant performance improvements for a larger class of applications. An example of the success of such integration is somehow provided by compressive classification [24]. In the future, we aim at further expanding this knowledge for other video and image analysis tasks, focussing in particular on increasing the coding efficiency with respect to traditional disjoint approaches. As

for the specific case of the tracking applications considered in this paper, we are currently working on refining the prior information used for the weighted CS reconstruction, and on including more sophisticated background subtraction techniques.

REFERENCES

- [1] R.G. Baraniuk, "Compressive Sensing," *Signal Processing Magazine, IEEE*, vol. 24, no. 4, pp. 118–121, 2007.
- [2] E. Candes, "Compressive sampling," in *International Congress of Mathematicians*, Madrid, Spain, 2006.
- [3] D.L. Donoho, "Compressed Sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.
- [4] M.B Wakin, J.N. Laska, M. F. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. F. Kelly, and R. G. Baraniuk, "An architecture for compressive imaging," *IEEE International Conference on Image Processing*, pp. 1273 – 1276, 2006.
- [5] L. Jacques, P. Vandergheynst, A. Bibet, V. Majidzadeh, A. Schmid, and Y. Leblebici, "CMOS Compressed Imaging by Random Convolution," Tech. Rep., Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, 2008.
- [6] I. Drori, "Compressed video sensing," in *BMVA Symposium on 3D Video - Analysis, Display, and Applications*, 2008.
- [7] Stankovic V., Stankovic L., and Cheng S., "Compressive video sampling," in *Proc. Eusipco-2008 16th European Signal Processing Conference*, Lausanne, Switzerland, August 2008.
- [8] J.Y. Park and M.B. Wakin, "A multiscale framework for compressive sensing of video," in *Proc. of Picture Coding Symposium*, Chicago, USA, May 2009.
- [9] M. Tagliasacchi, G. Valenzise, and S. Tubaro, "Hash-based identification of sparse image tampering," *IEEE Trans. Image Process.*, 2008, Submitted.
- [10] V. Cevher, A. Sankaranarayanan, M.F. Duarte, D. Reddy, R.G. Baraniuk, and R. Chellappa, "Compressive sensing for background subtraction," in *Proc. European Conf. on Computer Vision*, Marseille, France, October 2008.
- [11] E. Candes, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics*, vol. 59, pp. 1207 – 1223, 2006.
- [12] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.
- [13] E. van den Berg and M. P. Friedlander, "In pursuit of a root," Tech. Rep. TR-2007-19, Department of Computer Science, University of British Columbia, June 2007, Preprint available at http://www.optimization-online.org/DB_HTML/2007/06/1708.html.
- [14] E. Candes, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *Journal of Fourier Analysis and Applications, special issue on sparsity*, 2008.
- [15] A. Gersho and R.M. Gray, *Vector Quantization and Signal Compression*, Springer, 1992.
- [16] M. Cossalter, G. Valenzise, and S. Tagliasacchi, "Privacy-enabled object tracking in video sequences using compressive sensing," in *Proc. Int. Conf. Automatic Video and Signal-Based Video Surveillance*, Genoa, Italy, September 2009.
- [17] M.B Wakin, J.N. Laska, M. F. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. F. Kelly, and R. G. Baraniuk, "Compressive imaging for video representation and coding," in *Proc. of Picture Coding Symposium (PCS)*, Beijing, China, 2006.
- [18] M. Piccardi, "Background subtraction techniques: a review," in *IEEE International Conference on Systems, Man and Cybernetics*, 2004, pp. 3099 – 3104.
- [19] Andrea Bonarini, Matteo Matteucci, Davide Migliore, and Matteo Naccari, "A robust approach to motion detection and tracking in indoor video surveillance," 2005.
- [20] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: particle filters for tracking applications*, Artech House, 2004.

- [21] D. Rowe, I. Rius, J. Gonzles, and J. J. Villanueva, “Robust particle filtering for object tracking,” *International Conference on Image Analysis and Processing*, vol. 3617, pp. 1158–1165, 2005.
- [22] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Prentice Hall, 2007.
- [23] D. Marpe, H. Schwarz, and T. Wiegand, “Context-based adaptive binary arithmetic coding in the h.264/avc video compression standard,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. 7, pp. 620–636, 2003.
- [24] Mark A. Davenport, Marco F. Duarte, Michael B. Wakin, Jason N. Laska, Dharmpal Takhar, Kevin F. Kelly, and Richard G. Baraniuk, “The smashed filter for compressive classification and target recognition,” in *Proc. SPIE Symposium on Electronic Imaging: Computational Imaging*, 2007, p. 6498.