Online Space-Variant Background Modeling With Sparse Coding

Alessandra Staglianò, Nicoletta Noceti, Alessandro Verri, and Francesca Odone

***Abstract*— In this paper, we propose a sparse coding approach to background modeling. The obtained model is based on dictionaries which we learn and keep up to date as new data are provided by a video camera. We observe that, without dynamic events, video frames may be seen as noisy data belonging to the background. Over time, such background is subject to local and global changes due to variable illumination conditions, camera jitter, stable scene changes, and intermittent motion of background objects. To capture the locality of some changes, we propose a space-variant analysis where we learn a dictionary of atoms for each image patch, the size of which depends on the background variability. At run time, each patch is represented by a linear combination of the atoms learnt online. A change is detected when the atoms are not sufficient to provide an appropriate representation, and stable changes over time trigger an update of the current dictionary. Even if the overall procedure is carried out at a coarse level, a pixel-wise segmentation can be obtained by comparing the atoms with the patch corresponding to the dynamic event. Experiments on benchmarks indicate that the proposed method achieves very good performances on a variety of scenarios. An assessment on long video streams confirms our method incorporates periodical changes, as the ones caused by variations in natural illumination. The model, fully data driven, is suitable as a main component of a change detection system.**

***Index Terms*— Background subtraction, space-variant, representations, dictionary learning and sparse coding.**

I. Introduction

MODELING the background of a scene viewed by a fixed camera is a classic problem of computer vision. Its effective solution is an essential step for addressing subsequent problems, like tracking and recognition of moving objects [37], [50]–[52], [57] and dynamic scene understanding [3], [24], [36], [38], [56], [62], common in video surveillance [7], [11], [21], [29], [33], human-computer interaction [19], [39], [45], [58], and industrial applications [13], [17], [60]. Aside from the unavoidable noise affecting image acquisition, several factors contribute to make this problem still challenging: among the most important we mention *illumination changes* - no matter whether happening smoothly or abruptly over time, *intermittent image variation induced by objects belonging to the background* like foliage or poles shaken by the wind in an outdoor scene, and *permanent background changes* like a door which may be open or closed in an indoor environment.

In building a background model for a scene viewed by a fixed camera a method can rely on some specificities. First, the background - or a great part of it, rarely changes over time. When it does change, the new background remains stable, at least for a while. A consequence of the background temporal stability is that the observed image sequence can be thought of as a stream of weakly supervised noisy input data. In this work we model the background by making use of dictionary learning techniques inducing sparse representations [28]. This choice allows us to leverage on the specificities of above. The image is divided into (possibly overlapping) patches of equal size and each patch is processed individually in order to exploit the changes locality typical of the application domain, and to allow for high parallelization. In a bootstrap phase, the dictionary is simply derived by the first few frames of the video sequence. At run time our method approximates a patch with a sparse linear combination of atoms (forming the dictionary) learnt online from the image stream. If the reconstruction error is high a change, or anomaly, is detected in the image patch. Long lasting anomalies call for a dictionary update. The fact that many consecutive very similar image patches are fed as examples into the dictionary learning module ensures that, for each patch, the dictionary grows adaptively in size and only when the reconstruction accuracy of the current image patch is not sufficient. By looking for sparse reconstruction, not only the dictionary size is kept under control, but the obtained reconstruction can also be computed and maintained effectively. Image variations not leading to stable changes do not trigger an update of the current dictionary and are discarded. The proposed online learning procedure can be viewed as a core of a change detection system able to automatically adapt to the environment evolution keeping track of the relevant background variations without the need of human intervention.

A. Background Modeling Methods

The literature of background modeling is rather wide and a complete overview is out of the scopes of this paper – the interested reader is directed to [6] and [43] for complete surveys on the subject. Here we just summarize the main contributions to the topic. A common choice is to model each pixel of the background independently. Very popular unimodal approaches, as the (weighted) running average, filter and gaussian-based methods [1], [9], [57] lie on this category.

Manuscript received March 11, 2014; revised August 17, 2014 and January 2, 2015; accepted March 19, 2015. Date of publication April 9, 2015; date of current version April 29, 2015. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Sylvain Paris.

The authors are with the Dipartimento di Informatica, Bioingegneria, Robotica e Ingegneria dei Sistemi, Università degli Studi di Genova, Genoa 16146, Italy (e-mail: alessandra.stagliano@unige.it; nicoletta.noceti@unige.it; alessandro.verri@unige.it; francesca.odone@unige.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2015.2421435

Probabilistic methods are often based on the temporal prediction of the pixel evolution over time by means of filter – e.g. Wiener of Kalman filters [35], [54]. Despite their computational efficiency, all these methods let the successive post-processing in charge of considering the spatial consistency of the results.

Methods that operate on blocks rather than single pixels have also been proposed. In [46] the authors assume that neighboring blocks of background pixels follow similar variations over time, and train a Principal Component Analysis to model each block. However, such technique lacks a mechanism to adapt the models over time. A two-level method has been adopted in [30], which detects background image blocks by means of classification, and then exploits the obtained outcomes to perform block-wise updates.

The work in [8] is an attempt to cast the background subtraction task in the framework of compressive sensing. A major drawback is that it needs the foreground object to be of relatively small size with respect to the camera view. In [32] instead, the authors are inspired by the biological mechanisms of motion-based perceptual grouping, and propose a solution that well adapts to highly dynamic scenes, but requires very high processing time.

In highly complex and challenging environments, where the use of a single value or model for each pixel leads to inaccurate results, a viable solution is provided by the so-called multimodal background models. Perhaps the most popular model belonging to this category is the Gaussian Mixture Model and variants (see [41], [50]). The major drawbacks of the method reside in the strong assumption that the background is visible most of the time and is characterized by a low variance. Parameters selection may also be very complex. To avoid the choice of a specific probability density function, non-parametric models have also been proposed, as the ones based on kernel density estimation [16], [42]. These methods are based on building a histogram of background values accumulated on the recent pixel history. However, they often fail to cope with concomitant events evolving at different speeds, and strongly depend on an appropriate choice of the time interval.

An alternative relies on using codebooks models [25]. Each pixel is described via a certain number of codewords, each one representing a particular color configuration when part of the background. Improvements aiming at incorporating spatial and temporal context of the pixel have been proposed [4], [59]. Layered representations are an interesting alternative to address the variability of a background [40].

B. Background Modeling With Sparse Coding

More relevant to our work are previous approaches tackling background modeling as a sparse recovery problem [14], [15], [20], [22], [47], [63]. Often times, the problem formulation relies on the definition of the foreground as the error between an image frame and its approximation – obtained as a linear combination of a fixed set of frames – and on the attempt of sparsifying the contributions to such error – leveraging on the assumption that most of the image locations correspond to background. The works in [14] and [15] represent the first attempts in this direction, the former building the linear approximation on the basis of a fixed coding matrix obtained in a training phase as concatenation of images, the latter comparing different methods to select a dictionary. More recently the sparse coding step has been improved [12], [23], [47], [63] and naif procedures for dictionary update has been proposed [23], [47].

C. Contributions

Our method represents a joining link between block-based approaches, to account for spatial consistency, and multi-modal methods, to deal with arbitrary complex scenarios and dynamic events.

Although inspired by similar considerations, our work significantly diverges from previous applications of sparse coding and dictionary learning to background modeling and foreground detection [12], [14], [15], [23], [47], [63]. First, *we aim at sparsifying the patch representation rather than the error contributions*, leading to a different mathematical formulation of the problem. Second, our method is formulated as *a set of many small-size optimization problems* – one for each patch – instead of as a single global problem defined on data of far higher dimensionality (size of a patch as opposed to size of the entire image). This provides the method with *local adaptiveness*, a relevant property in many surveillance scenarios. Third, the update procedure is designed to be activated only *when and where* it is necessary. Each dictionary size reflects the amount of dynamics of the corresponding patch, since the model controls the redundancy and is able to adapt to different complexities. As a consequence, the spatial demand is not uniform on all image locations, but is larger when a richer set of configurations need to be captured while it is very small on more stable regions.

Related to this aspect is the computation of the pixel level of details, achieved by a more traditional pixel-based background subtraction and only applied to the image patches that changed with respect to their background model. Thanks to this coarse-to-fine approach we are able to control the computational cost of the method, while achieving accurate results at a pixel precision whenever needed.

In an early version of the paper [49] we presented the core idea of the proposed algorithm (reported here in section III-A). In this paper we set the methodology underpinning, present the full computational pipeline and provide a complexity analysis. We also assess the method on two benchmarks (the SABS dataset and the Change Detection dataset) to evaluate its appropriateness on different types of scenarios and motion and to compare it with state of the art. More importantly, we analyze the results obtained by the method across long time observations: we process a video stream acquired in-house covering two long periods of time acquired on two consecutive days and show how our method is able to learn illumination patterns that periodically occur at the same time of the day. Similar considerations could be made on other types of stable scene changes and intermittent motion patterns.

The reminder of the paper is organized as follows. Section II summarizes the concepts behind dictionary learning and

reports an overview of the algorithmic aspects of the method; Section III describes the Background Modeling Through Dictionary Learning (BMTDL) we propose, while Section IV reports an analysis of the method on a few meaningful study cases. Section V is about the method assessment on the two benchmark datasets, while Section VI shows the method performances when long time observations are available. Finally VII is left to a final discussion.

II. Learning Parsimonious Data Representations

In this section the idea behind the concept of sparse dictionary learning is briefly reviewed along with the main algorithmic issues arising in the optimization stage.

Sparse dictionary learning (DL) aims at building data representations by decomposing each datum into a linear combination of a few components selected from a dictionary of basic elements, called *atoms*. The adaptiveness of the dictionary is achieved through the process of learning the atoms directly from the input data instead of using a fixed, over-complete dictionary derived analytically (as for Wavelets or Discrete Cosine Transform [34]).

A. ℓ_1-Dictionary Learning

We consider a training set of elements $\mathbf{x}_i \quad i = 1, \ldots, N$, with $\mathbf{x}_i \in \mathbb{R}^n$, and we assume the existence of a dictionary of K atoms $\{\mathbf{d}_1, \ldots, \mathbf{d}_K\}$ such that $\mathbf{x} \approx \mathbf{D}\mathbf{u}$ for some sparse K-vector $\mathbf{u}$ and where $\mathbf{d}_j$ is the $j-th$ column of the $n \times K$-matrix D. The dictionary is learnt from data, together with the coding of the training set. The problem can be formulated as the following optimization

$$\min_{\mathbf{D},\mathbf{u}} \|\mathbf{x} - \mathbf{D}\mathbf{u}\|_2^2 + \lambda \mathrm{R}(\mathbf{u}) \quad \textit{subject to} \quad C(\mathbf{u}) \tag{1}$$

where R is an appropriate regularization term and C is a set of constraints that can be added to the optimization (different choices can be found in [26], [28], [55], and [61]). The first term is a data-fidelity term that ensures a small reconstruction error over the training set, while the second is a regularization term. The trade-off between the two terms is obtained by appropriately tuning the positive regularization parameter λ. In this work we refer in particular to sparse coding, also known as ℓ_1-Dictionary Learning [28]:

$$\min_{\mathbf{D},\mathbf{u}} \|\mathbf{x} - \mathbf{D}\mathbf{u}\|_2^2 + \lambda \|\mathbf{u}\|_1 \quad \textit{subject to} \quad \|\mathbf{d}_j\|_2 \leq 1. \tag{2}$$

In this case, the regularizer is a penalty term inducing sparsity in the components of the $\mathbf{u}$ vector [53].

Although the joint minimization problem in $(\mathbf{D}, \mathbf{u})$ is non-convex and non-differentiable, it is convex in each variable and it can be solved by iteratively minimizing first with respect to $\mathbf{u}$ (*sparse coding step*) and then to $\mathbf{D}$ (*dictionary update step*), assorting to a procedure known as *block-coordinate descent* [28]. We now briefly summarize the procedure, where each minimization is solved by proximal methods, referring the interested reader to [5].

1) Sparse Coding: With fixed $\mathbf{D}$, the functional (2) takes the form $E(\mathbf{U}) = F(\mathbf{U}) + J(\mathbf{U})$, with $\mathbf{U}$ a $K \times N$ matrix of all the codes. The two components are expressed by

$$F(\mathbf{U}) = \frac{1}{n}\|\mathbf{X} - \mathbf{D}\mathbf{U}\|_F^2 \quad \text{and} \quad J(\mathbf{U}) = \frac{2\lambda}{K}\sum_{i=1}^{N} \|\mathbf{u}_i\|_1. \tag{3}$$

where $\mathbf{X}$ is the $n \times N$ data matrix, whose columns are the training vectors $\mathbf{x}_i$, $i = 1 \ldots N$. J is a non smooth term. We can minimize the functional E with a proximal algorithm obtained combining a projection step with a forward gradient descent step, obtaining the following update rule:

$$\mathbf{U}^p = \mathbf{S}_{\lambda/K\sigma_U}\left[\mathbf{U}^{p-1} - \frac{1}{2\sigma_U}\nabla_{\mathbf{U}} F(\mathbf{U}^{\mathbf{p-1}})\right] \tag{4}$$

where

$$\nabla_{\mathbf{U}} F = -\frac{2}{n}\mathbf{D}^T(\mathbf{X} - \mathbf{D}\mathbf{U}) \tag{5}$$

is the gradient of the (strictly convex) differentiable term F, while

$$(\mathbf{S}_\lambda[\mathbf{U}])_{ij} = \mathrm{sign}(\mathbf{u}_{ij}) \max\{|\mathbf{u}_{ij}| - \lambda, 0\} \tag{6}$$

is the component-wise soft-thresholding operator corresponding to the proximity operator of the term J.

2) Dictionary Update: When $\mathbf{U}$ is fixed the quadratic constraint on the columns of $\mathbf{D}$ is equivalent to an indicator function J. Denoting by $\pi(\mathbf{d}) = \mathbf{d}/\max\{\mathbf{1}, \|\mathbf{d}\|\}$ the projection on the unit ball in $\mathbb{R}^d$, let π_D be the operator applying π to the columns of $\mathbf{D}$. In this case the update step is

$$\mathbf{D}^p = \pi_D(\mathbf{D}^{p-1} + \frac{1}{\sigma_D}(\mathbf{X} - \mathbf{D}^{p-1}\mathbf{U})\mathbf{U}^T). \tag{7}$$

3) Iterative Gradient Descent: σ_U and σ_D are the gradient descent step sizes and control the convergence rate. In general one can choose a step size to be equal to the Lipschitz constant of ∇_F, although faster convergence rates may be obtained by choosing adaptive step sizes or by modifying the proximal step (see [5] and references therein).

The initialization of $\mathbf{D}$ and $\mathbf{U}$ will be discussed in the next section for the specific application we are interested in. The iterations stop either upon reaching a maximum number of iterations fixed a priori or if the functional value changes by less than a fixed tolerance.

B. Dictionary Learning for Background Modeling

We now discuss how to apply dictionary learning to the problem of adaptively modeling the background of a scene. In this work we assume our data are image patches of a fixed size acquired by a fixed camera at different time instants. The choice is motivated by different considerations. A pixel-based modeling of the background, as the one proposed in [25] and [50] does not appear to be informative enough for learning a dictionary. Conversely methods based on learning a dictionary of the entire image [14], [15], [47], [63] do not seem to exploit the peculiarities of the application domain, where changes are often local and usually affect well defined portions of the image plane. Also, different portions of the image may

need models of different complexity, or may require more frequent updates. In this case, it might be an overkill to update the whole frame-model every time a small local background change occurs. Finally, a patch-based model allows us to choose an appropriate scale for the model (related to the patch size) and, in principle, to exploit prior information by selecting an appropriate size for a given position on the frame.

In what follows, each patch will be named $\mathbf{p}_{xy}^t$, meaning that the area we are considering is located at the position xy of the time instant t. A patch evolving in time will participate to form and update a specific dictionary $\mathbf{D}_{xy}$.

A peculiarity of the application domain is that data are provided as a (video) stream, one at a time. Therefore, our data are a sequence of $\mathbf{x}_i$ examples, $i = 1, \ldots$ – each one being the patch unfolded in a n-dimensional vector – where the observation at time t could possibly update the previous solution. In the machine learning terminology this would correspond to an *online learning* procedure [31]. Notice that the algorithm described in the previous section naturally applies to this setting, in Section III we will provide the details of the online procedure.

We conclude by observing that, if most training examples are very similar as in the case of an image sequence obtained from a fixed camera looking at a slowly changing background, the atoms correspond to denoised versions, or prototypes, of the inputs [2], [48]. Over time, a richer dictionary is expected to arise in order to be able to capture greater (and permanent) changes in the background as seen from the given image patch. Sparsity, which has been reported to favor discriminative power in subsequent classification tasks [5], [28], [44], here ensures a model of the background consisting of a linear combination of a few prototypes. The prototypes actually used in the model provide information which, in subsequent stages, can be usefully employed to reason on the time evolution of the viewed patch. The parameter K controls the number of learnt atoms: in our case it is typically small because relevant background changes are expected to be rare.

III. The BMTDL Method

In this section we summarize our method for Background Modeling Through Dictionary Learning (henceforth BMTDL). As a first thing, we divide each processed video frame in a regular grid of (possibly overlapping) patches. The whole processing scheme is carried out on each patch individually, allowing for high parallelization. In what follows for clarity, unless otherwise stated, we will concentrate on a single patch. We assume we are given a sequence of patches $\{\mathbf{p}_{xy}^t\}_t$ which may be seen as noisy versions of the same information if no dynamic events occur within the patch. All these elements may be approximated as linear combinations of the same dictionary $\mathbf{D}_{xy}$ built over time, and updated only when necessary. We then assume this dictionary may allow us to reconstruct different realizations of $\mathbf{p}_{xy}$ in a normal configuration (that is, when nothing significant occurs in the patch).

At run time t, the BMTDL method approximates the patch $\mathbf{p}_{xy}^t$ w.r.t. $\mathbf{D}_{xy}$. If the reconstruction error is small, this means the current patch can be accurately approximated with the available dictionary, therefore the patch is likely to be a background patch. Otherwise an anomaly is detected, possibly caused by the presence of a foreground object. If such anomaly persists for some time, it is likely to be a permanent background change, then the dictionary needs to be updated to accommodate new stable information.

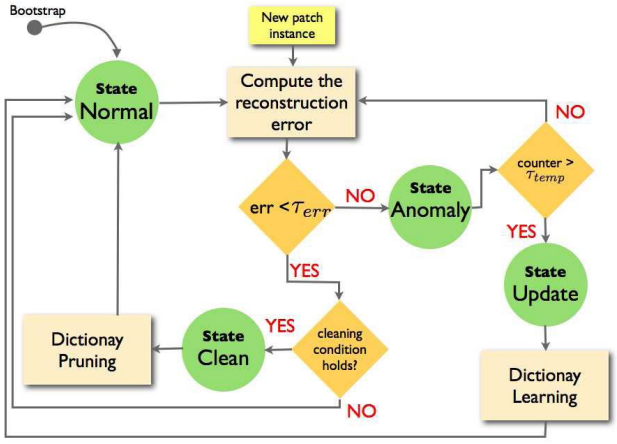


Fig. 1. The structure of the proposed system.

The method can be graphically described as in Fig. 1. In essence, each patch can be in one among 4 possible states. Two of them – states NORMAL and ANOMALY – are (possibly) *persistent*, since they refer to, respectively, the situations when nothing is happening or a foreground object is altering the appearance of the patch. The remaining two are *transient* states, and refer to UPDATE and CLEAN operations applied to keep the background model up-to-date while controlling its size.

In the remainder of the section, each part of the method is described in a greater detail.

A. Run Time Analysis

On a bootstrap phase, each dictionary $\mathbf{D}_{xy}$ is initialized with the first k patches normalized by their average in order to constrain the norm of the atoms to be no greater than 1: $\mathbf{D}_{xy} = \{\mathbf{p}_{xy}^t / \overline{\mathbf{p}}\}_{t=0}^{k-1}$ with $\overline{\mathbf{p}} = \frac{1}{k}\sum_{t=0}^{k-1} \mathbf{p}_{xy}^t$. Notice it is not important the initial dictionary is meaningful since the run time procedure described below will take care of noise or moving objects included in the initial dictionary. We now consider each possible state of the system in a given time instant t, following the bootstrap (see Fig. 1).

1) NORMAL: at run time, at each instant t the input patch $\mathbf{p}_{xy}^t$ is decomposed with respect to the current dictionary $\mathbf{D}_{xy}$. To do so we compute a feature vector $\mathbf{u}$ as $\mathbf{p}_{xy}^t \approx \mathbf{D}_{xy}\mathbf{u}$ by minimizing Eq. 2 with respect to $\mathbf{u}$ only. This can be done by applying the iterative gradient descent with the update rule (4). At each t $\mathbf{u}$ can be initialized with the output of instant $t-1$, thus we obtain a very fast convergence. We then estimate the reconstruction error

$$||\mathbf{p}_{xy}^t - \mathbf{D}_{xy}\mathbf{u}||_2 / ||\mathbf{p}_{xy}^t||_2.$$

If the reconstruction error is lower than a threshold τ_{err} the patch remains in a NORMAL status; the system finds the $\mathbf{u}$ non-zero coefficients and increases a *usage counter* assigned to

(a) (b) (c)

Fig. 2. The three steps needed to derive a pixel-precision binary map from the patch-precision map: (a) The coarse binary map at the patch level (b) The absolute difference between the current patch and the most similar background atom (c) The binary map at piexl precision obtained by thresholding.

their corresponding atoms. These counters represent a measure of how many times an atom has been used. They will be needed in a subsequent step to discard low-rated atoms from the dictionary, to control its space occupancy.

2) ANOMALY: If the reconstruction error is greater than the threshold τ_{err} an anomaly is detected and the system changes status. A counter measuring the persistence of the change is increased.

3) UPDATE: If the system remains in the state ANOMALY for τ_{temp} frames, it means that a stable background change occurred in the scene. Using the last τ_{temp} frames as training data, we learn a new dictionary $\mathbf{D}_{xy}^{update}$ of a fixed size K_{update} by minimizing Eq. 2, and add the new estimated atoms to the main dictionary $\mathbf{D}_{xy} = \mathbf{D}_{xy} \cup \mathbf{D}_{xy}^{update}$. Then the system returns in the state NORMAL and begins to process new patches with the updated dictionary. This step provides a good adaptation of the system to new background configurations without losing relevant information about previous configurations of the scene. This allows us to obtain a background model with a memory, so that if some event happens with a certain frequency, with the exception of the first time, it will not be classified as foreground. The value of τ_{temp} is influenced by the stationarity property of the background under analysis and in turn influences its capability of adaptation in time [54]. It turns out that it is strongly dependent on the specific scene and its dynamics, thus these aspects should be considered in the selection process (similar considerations may be found in [4], [6], [7], and [25]).

The size of the update, K_{update}, should be chosen to control the overall size of the dictionary but, at the same time, allowing for variations of different complexity.

4) CLEAN: In order to control the size of the model, that may significantly grow as a consequence of different update steps, and discard atoms which are not useful or even wrong a cleaning step is required.

Every τ_B frames, the system checks the usage counters associated with the recently added atoms and discards the ones which have seldom been used (whose counter is below a threshold which may depend on the expected duration of foreground dynamic events). Doing so, the part of the dictionary that is actually checked is the one that has been updated last, that might contain atoms produced by a leakage of foreground information. In this case it is reasonable to assume the corresponding atoms will be rarely (or never) used in the future. Once a portion of the dictionary has been cleaned, it can be considered as a consolidated part of the background history.

B. Foreground Detection

The state **ANOMALY** corresponds to a variation with respect to the available dictionary which could be caused by a foreground moving object. At a given time t all patches in the state **ANOMALY** form a change detection map with a patch precision (see Fig. 2 (a)). If a pixel-precision foreground map is needed, a further step is required. To this purpose, when a patch $\mathbf{p}_{xy}^t$ is marked as anomalous, we look for the nearest neighbor atom in its dictionary with respect to the euclidean distance, be it $\mathbf{d}_{xy}^{bg}$.

We then compute the absolute difference between the normalized $\mathbf{p}_{xy}^t$ and $\mathbf{d}_{xy}^{bg}$. If patches are not overlapping the union of all absolute differences gives us an overall absolute difference image. In presence of overlap, the different contributions are summed up to compute the final difference image. This leads to image areas which are more likely to be foreground, corresponding to locations where more than a patch presents a high reconstruction error. The difference image is finally binarized with an appropriate threshold.

C. Computational Costs

We now briefly discuss the computational load of our method to process a patch. At each time instant, representing the patch $\mathbf{p} \in \mathbb{R}^n$ with respect to the dictionary $\mathbf{D} \in \mathbb{R}^{n \times K}$ requires an iterative procedure of p steps. Each step includes two main operations, i.e. the computation of the gradient (Eq. 5, cost in $O(Kn)$), and the soft-thresholding, performed according to Eq. 6 (cost in $O(K)$). For similar considerations, this is also the cost of dictionary updating. The computation of the reconstruction error is for free, being included in the iterative procedure. Hence, each step costs $O(nK)$, leading to a temporal complexity of $O(nK) \times p$ steps for the computation of the binary map at a patch level.

The additional cost required to reach a pixel-level precision – obtained after the comparison of the current patch with all the atoms in its dictionary – is again proportional to $O(nK)$. Notice, however, this overload is paid only for patches marked as anomalous during the coarse procedure, only a small percentage of the entire image on average.

For comparison, let us consider the temporal complexity at a patch level, of one of the most efficient methods for building and maintaining a background model, the running average

TABLE I
F-MEASURE ON A VALIDATION SET TAKEN FROM THE "BASELINE" GROUP (CHANGE DETECTION DATASET) FOR DIFFERENT VALUES OF THE REGULARIZATION PARAMETER. A HIGHER λ CORRESPONDS TO A SPARSER SOLUTION

λ	0.001	0.010	0.050	**0.100**	0.300	0.500
F-measure	0.863	0.864	0.864	0.865	0.863	0.810
used atoms (%)	99.9	98.8	86.9	85.9	79.8	78.7

algorithm [57]. Running average first relies on detecting the locations within the patch in which some change occurred. Then, the method updates the background as a weighted sum of previous background and current image, that in the worst case will affect the entire patch. Finally, the binary map is obtained by thresholding the absolute difference between current image and current background. All these operations have a cost linear in the patch size, $O(n)$. Thus, the difference between the two methods depends on the size of the dictionary and on the number of iterations required. The latter are usually small, since we initialize $\mathbf{u}_t$ with $\mathbf{u}_{t-1}$. The former depends on the dynamics of the patch, but it is usually small and always $K \ll n$.

On the matter of temporal complexity a final remark is in order. The computational cost our method requires to process the entire image (approximately of order $P(O(n\bar{K}) \times \bar{p})$, with P the number of patches, $\bar{K}$ and $\bar{p}$ the average number of atoms and iterations over all the patches) is comparable to the cost we would pay for applying our method to the estimation of a single global dictionary, with a single patch of the size of the entire image. In that case, given the dimensionality of the input data equal to the full image size $m = Pn$, the complexity is of order $O(mK) \times p$, where K is the cardinality of the single global dictionary while p is the number of iterations. From a theoretical standpoint, this is in line with the principle of domain decomposition [18], [27].

IV. THE METHOD AT WORK

In this section we analyze a few study cases, focusing in particular on uniform and non uniform illumination changes, and stable background variations.

Since our goal is to approximate background patches, obtaining at the same time a high reconstruction error for foreground elements, our choice is restricted to sparse models. We experimentally observed that other data-driven dictionaries (e.g. obtained with K-Means) perform comparably, in general, to ℓ_1- dictionary learning, even though the latter is slightly more accurate. The use of more naive models (e.g. dictionaries obtained by randomly picking the atoms from the observed patches) instead, fail to cope with more challenging scenarios. We experimentally observe the choice of the regularization parameter λ is not critical for the overall procedure. In Table I we report the F-measure obtained by varying λ while processing a validation set (taken from the Change Detection dataset — see Section V-B). Therefore, in all the experiments presented in the paper λ will always be fixed ($\lambda = 0.1$), chosen as a compromise between sparsity and accuracy.

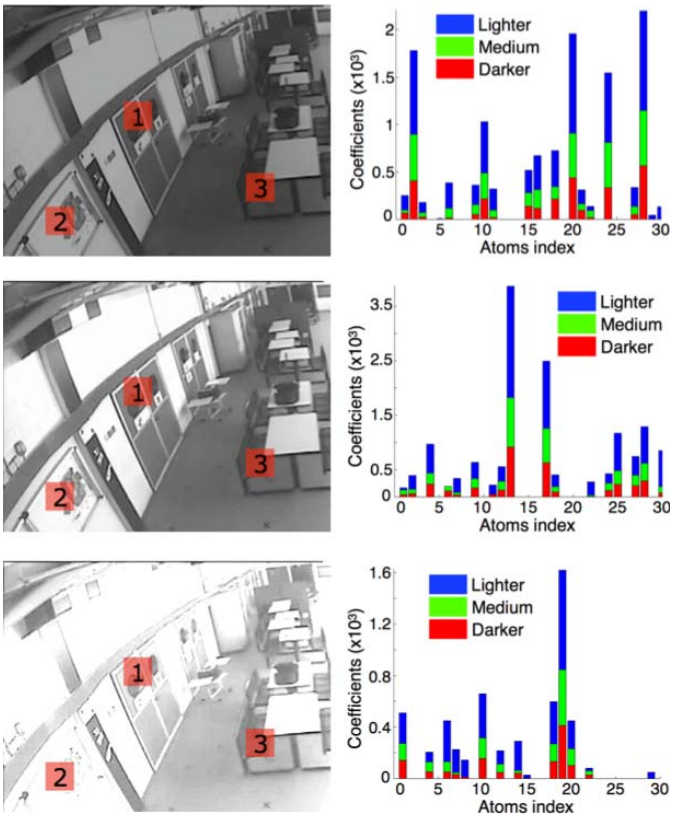


Fig. 3. An analysis of the coefficients used for the reconstruction of three images, under different levels of illumination and using a fixed dictionary. Left column: three frames corresponding to different illumination conditions (from top: darker, medium, lighter). Right column: bar plots of the $\mathbf{u}$ vectors for the three patches (from top: patch 1, 2, 3) under the three different illuminations show consistent variations on the coefficients.

As for the patch size, in this section we set it to 50×50 pixels. The level of overlapping has been set to 10 pixels.

In the reminder of the section we will discuss in details how the proposed method behaves on a selection of different challenging circumstances.

A. Uniform Illumination Changes

We discuss here the robustness of the dictionary-based background model with respect to global illumination changes. A global uniform change on a patch can be formulated as the product of the pixels value within the patch by a constant. In our background model, this change will not affect the dictionary but will change the coefficients $\mathbf{u}$ of the linear combination. This observation conveys a very powerful concept of our model: the same dictionary can be exploited to model backgrounds characterized by different levels of illumination by simply appropriately selecting the coefficients.

To support this assertion, it is important to check whether we can achieve, under these circumstances, comparable reconstruction errors. In Fig. 3, left column, we show three video frames of the same environment under different illuminations. For the sake of this analysis, we sample three patches from the image – highlighted in the figure – and learn dictionaries the size of which was set a priori ($K = 30$) from a set of

TABLE II
A TABLE REPORTING THE OVERALL RECONSTRUCTION ERROR OF THE THREE IMAGES IN FIG. 3 WITH RESPECT TO THE FIXED DICTIONARY. NOTICE THE STABILITY OF THE RECONSTRUCTION ERROR

	Reconstruction error		
patch #	medium	lighter	darker
1	0.133	0.135	0.156
2	0.114	0.109	0.116
3	0.219	0.207	0.222

images acquired under a "medium" illumination (similar to the central frame in Fig. 3). We represent the three frames with respect to the *same* dictionary and evaluate their reconstruction errors. The right column of Fig. 3 shows the results obtained on the three patches: each plot bar represents the coefficients needed to reconstruct the patches under the different light conditions. Notice how the coefficients change coherently on the various atoms: in particular, the coefficients used to reconstruct the image with lower illumination are consistently smaller than the ones used for the "medium" image. Similarly, the "lighter" image has been reconstructed with consistently higher coefficients. Table II summarizes the reconstruction error on the three patches: it is immediate to see that the values are stable as the illumination changes.

B. Background Stable Changes and Local Shadows

We have seen that our method does not require an update of the dictionary after a uniform change of the patch intensity. On the other hand, an uneven illumination can generate new apparent structures in the image (corresponding, for instance, to the edges of a shadow) and in this case we may need to add new atoms to the corresponding dictionary to account for them in the future. Fig. 4 shows a scene modified during the day by a blade of sun light projected onto the floor. On the left we show the atoms of the dictionary of one of the patches affected by the illumination change (highlighted on the frames). The atoms (whose insertion order should be read column-wise on the image) depict the main structural changes in the patch appearance. Notice that uniform changes have not been included in the dictionary, as we discussed before.

From the point of view of a background update the dynamic event we have just described is equivalent to a real structural change in the scene, caused, for instance, by objects added or moved in a new and permanent position. Fig. 5 shows instead an example of a video sequence containing short-term dynamic events (people walking in the scene) and a long-term variation (a door left open). In this analysis, for clarity, we consider very small dictionaries with $K = K_{update} = 1$. We focus on three different patches corresponding to the door region, on the left of the image we show the dictionaries corresponding to the three patches; then the figure contains the sample frames (top) and the change detection map with patch precision (bottom). The presence of the people which entered the scene is correctly detected at frame #1096. As the video evolves, the door is opening and the corresponding event is still captured by our method (#2153). At frame #3214, the variation in the background has became stable, and the

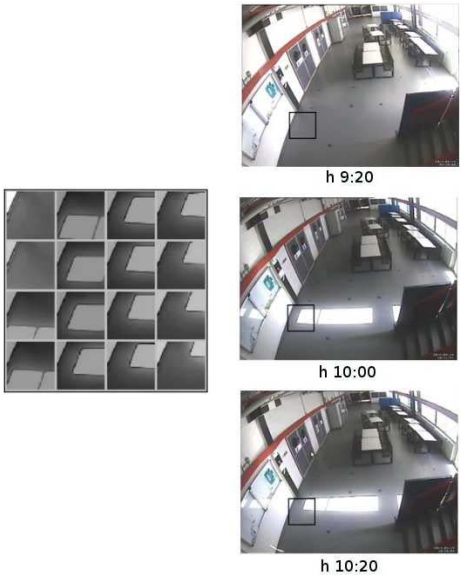


Fig. 4. A blade of sun light cast on the floor moving slowly in time. Left: a meaningful set of atoms of the dictionary associated with an unstable image patch at the end of the phenomenon (to be read column-wise). Right: three samples of the scene as the illumination change occurs — the patch used in the analysis is marked with the black square.

model is again able to correctly reconstruct the corresponding patches. As for the dictionary evolution, at frame #100 the dictionaries are composed by a single atom, computed in the bootstrap phase (the one on the left in each dictionary). While the action is occurring, the atoms are no longer able to provide an appropriate patch reconstruction, thus new atoms are added. In the final sets of atoms different configurations of the same patch can be identified. Notice that the central dictionary also contains an intermediate atom in which the door is not completely open. This is because the door was left in this state for some time and the appearance of this patch triggered a dictionary update. All the other temporary changes have not been included in the model.

Given the richness of the model we build as time goes by, if the "door opening" event occurs often, then the atoms involved will be often accessed and will always be available — reducing the amount of false alarms due to structural changes in the scene. This property can be exploited whenever we observe periodic changes, even frequent ones, as in camera jitter. In this case the various states will be stored in the background model and will not cause any change detection in the future. Obviously, the patches affected by the change will see their dictionary size growing accordingly. The idea we pursue has been addressed in the past with different techniques [4], [25], [30], [50]: the method we propose allows us to incorporate the variations in an elegant way, exploiting the theoretical soundness of the dictionary learning framework.

V. METHOD ASSESSMENT

In this section we report a quantitative experimental analysis of our method on two benchmark datasets, SABS and Change Detection. In both cases we follow the experimental protocol

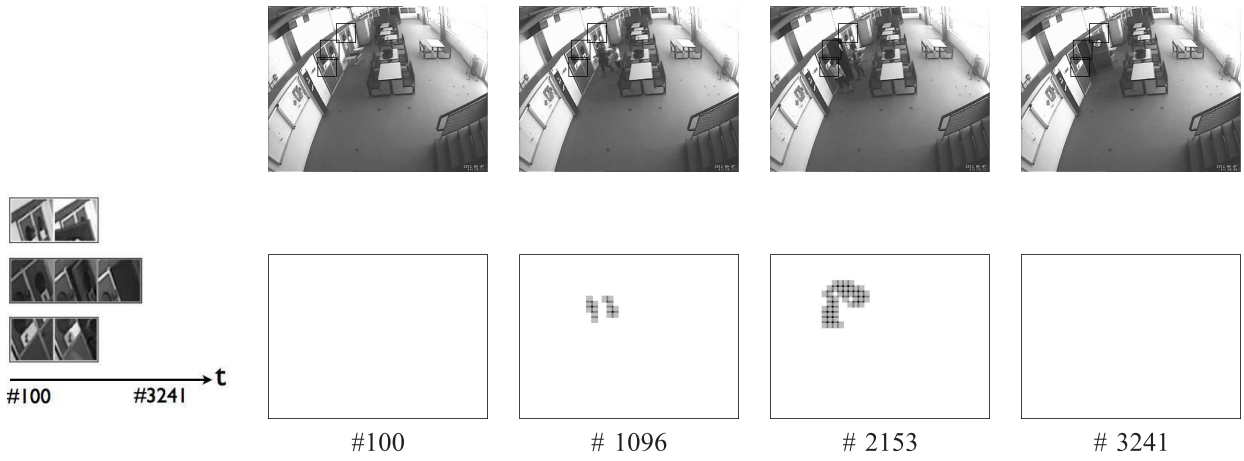


Fig. 5. An example of a stable change to be stored in the background model. On the right we report sample frames and their corresponding patch-based change detection maps: at Frame #100, the initial empty scene; Frame #1096, some people enter the scene; Frame #2153, a door is opened, inducing a stable background change; Frame #3241, the modified empty scene. Black squares mark patches we use as a reference in our discussion. On the left the three dictionaries corresponding to the patches highlighted, the numbers below refer to the frame index at which the atoms have been included in the models.

TABLE III

COMPARISON (F-MEASURES) ON THE SABS DATASET AS IN [7]. (1): BASIC SEQUENCES; (2) DYNAMIC BACKGROUND; (3) BOOTSTRAP; (4) DARKENING; (5) LIGHT SWITCH; (6) NOISY NIGHT; (7) CAMOUFLAGE; (8) NO CAMOUFLAGE); (9) H.264 COMPRESSION (40 kbps)

Method	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	Avg	Std.Dev.
Stauffer	**0.800**	0.704	0.642	0.404	0.217	0.194	0.802	0.826	0.761	0.594	0.255
Oliver	0.635	0.552	-	0.300	0.198	0.213	0.802	0.824	0.669	0.524*	0.255*
McKenna	0.522	0.415	0.301	0.484	0.306	0.098	0.624	0.656	0.492	0.433	0.176
Li	0.766	0.641	0.678	**0.704**	0.316	0.047	0.768	0.803	0.773	0.610	0.258
Kim	0.582	0.341	0.318	0.342	-	-	0.776	0.801	0.551	0.530*	0.205*
Zivkovic	0.768	0.704	0.632	0.620	0.300	0.321	**0.820**	**0.829**	0.748	0.638	0.199
Maddalena	0.766	0.715	0.495	0.663	0.213	0.263	0.793	0.811	0.772	0.610	0.232
Barnich	0.761	0.711	**0.685**	0.678	0.268	0.271	0.741	0.799	0.774	0.632	0.209
BMTDL	0.789	**0.736**	0.671	0.672	**0.329**	**0.591**	0.780	0.790	**0.778**	**0.681**	**0.149**

(*) average computed over fewer results

proposed by the authors of the benchmark, to compare our results with previously published one.

We first comment on parameter selection. As anticipated earlier we set the dictionary learning regularization parameter $\lambda = 0.1$. A critical parameter is the threshold on the reconstruction error, τ_{err}. When a training set was available we used the following procedure: we set $\tau_{err} = 0.3$ and processed the training frames. Then we corrected the τ_{err} based on the reconstruction errors obtained: we evaluated average error rec_{err} and its variance σ_{err}, then we took different values for τ_{err} lying in the range $[rec_{err} - \sigma_{err}, rec_{err} + 10\sigma_{err}]$. Usually the standard deviation is very small and the average reconstruction error is a good estimation of the threshold. We chose an asymmetric range to control the false positives. In absence of a training set, the same procedure was applied to the first part of the video. As for the parameters controlling the rate of update of the background model, we chose τ_{temp} in the range $[15'', 30'']$ depending on the video sequence, while τ_B is set to twice the value of τ_{temp}.

For a fair comparison with state-of-art methods, we produce binary maps at a pixel-level. This also allowed us to adopt the evaluation code provided with the benchmarks. The threshold on the image difference ranges from 30 to 60, depending on the set of videos under analysis.

A. The SABS Dataset

The SABS (Stuttgart Artificial Background Subtraction) dataset [7] (http://www.vis.uni-stuttgart.de/index.php?id=sabs) is an artificial dataset for pixel-wise evaluation of background models. It consists of video sequences representing different challenges in video surveillance, from dynamic elements in the background to sudden illumination changes. All the video sequences but one (*Bootstrap*) are provided with training frames that show the scene without any foreground object. The *Basic* sequence is a basic video-surveillance scenario; on a sub-sequence of it, it is possible to evaluate the system behaviour with respect to tree foliage. The *Darkening* sequence simulates a gradual illumination change in the scene, while the *Light Switch* shows several sudden illumination changes. The scene is acquired during the night, with some noise added, in the *Noisy Night* sequence. In the *Camouflage* sequence foreground and background have similar colors; as a comparison the dataset contains also a *NoCamouflage* sequence. Finally, the dataset offers acompressed version of the Basic sequence, *H.264*. We have fixed the patch size at

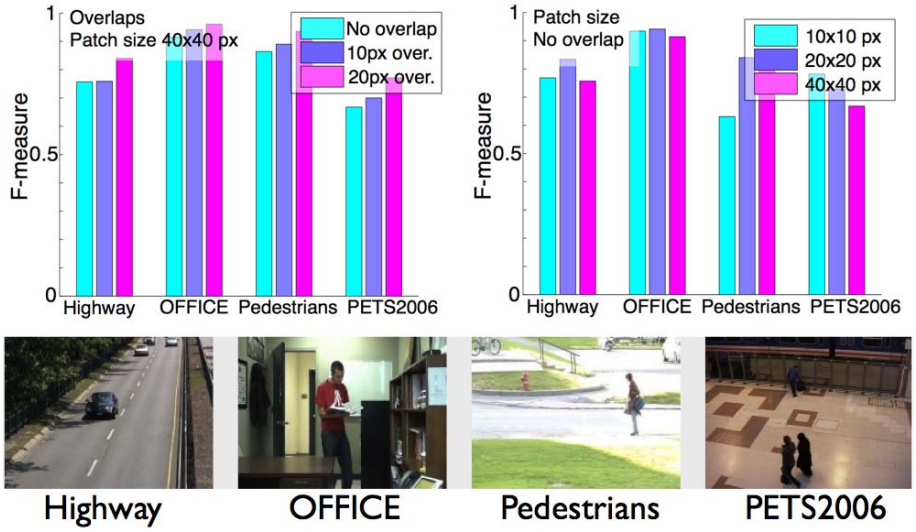


Fig. 6. The effect of varying the amount of overlap (left) and the patch size (right) on the F-measure obtained on the baseline videos (see text). Sample frames are reported below as a reference.

TABLE IV

DETAILED RESULTS OF BMTDL ACCORDING TO DIFFERENT PERFORMANCE MEASURES OVER THE WHOLE CHANGE DETECTION DATASET (FOLLOWING http://www.changedetection.net)

Cathegory	Recall	Specificity	FPR	FNR	PBC	FMeasure	Precision
Baseline	0.894	0.996	0.004	0.106	0.753	0.877	0.864
Dynamic BG	0.732	0.998	0.002	0.268	0.440	0.753	0.782
Camera Jitter	0.707	0.988	0.0123	0.293	2.496	0.725	0.748
Intermitt. Object Motion	0.684	0.982	0.016	0.316	4.421	0.686	0.709
Shadow	0.830	0.990	0.010	0.170	1.628	0.810	0.798
Thermal	0.862	0.980	0.020	0.138	2,790	0.793	0.737
OVERALL	0.785	0.989	0.011	0.215	2.088	0.774	0.773

TABLE V

BMTDL COMPARED (USING THE F-MEASURE) WITH THE MOST PROMISING METHODS (ACCORDING TO http://www.changedetection.net)

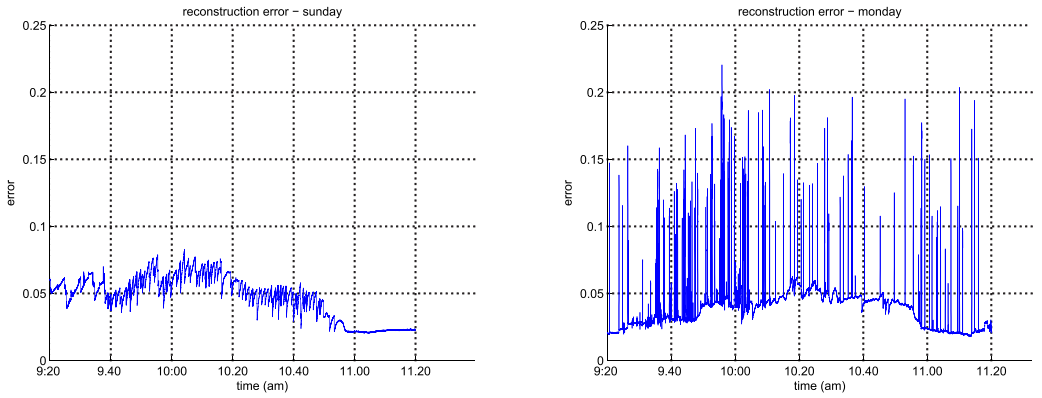
Method	Baseline	Dynamic BG	Camera Jitter	Intermittent OM	Shadow	Thermal	Avg.	Std.Dev.
DPGMM	0.93	0.81	0.75	0.54	0.81	0.81	0.78	0.13
BMTDL	0.88	0.75	0.72	0.69	0.81	0.79	0.77	0.07
SGMM SOD	0.92	0.68	0.70	0.70	0.87	0.71	0.76	0.10
PBAS	0.92	0.68	0.72	0.57	0.86	0.76	0.75	0.13
PSP MRF	0.93	0.70	0.75	0.56	0.79	0.69	0.74	0.12
SC SOBS	0.93	0.67	0.70	0.59	0.78	0.69	0.73	0.12
CDPS	0.92	0.75	0.49	0.74	0.81	0.66	0.73	0.15
SOBS	0.93	0.64	0.71	0.56	0.77	0.68	0.72	0.13
SGMM	0.86	0.64	0.73	0.54	0.79	0.65	0.70	0.12

40×40 pixels, then we have followed the experimental setup proposed by [7], and adopted the provided code for evaluating our results.

Table III extends the table reported on the dataset webpage to include our method. It can be noticed how the BMTDL method is very appropriate in most circumstances and achieves, on average, the best results, even if the video sequences of the SABS dataset are short (about 1000 frames) and do not leave much time to our method to properly adapt to the complexity and to the peculiarities of the scenarios.

B. The Change Detection Dataset

The Change Detection dataset (http://www.changedetection.net/) is a benchmark collection of 31 real-world videos acquired by standard and thermal video cameras and divided in six categories that include diverse motion and change detection challenges, both outdoor and indoor. The reported categories are *Baseline* (simple surveillance videos), *Dynamic Background* (with moving tree foliage, water and so on), *Camera Jitter* (acquired with slightly moving cameras), *Shadow* (with dark shadows), *Intermittent Object Motion* (with objects added to or removed from the scene) and *Thermal* (acquired with a thermal camera). All the videos were provided with training frames and in some of them the evaluation was made not on the whole frame but only in a region of interest. Here again we follow the experimental protocol provided by the authors of the benchmark.



(a) (b)

Fig. 7. A comparison of the trends of the average reconstruction error over sunday (a) and monday (b) morning.

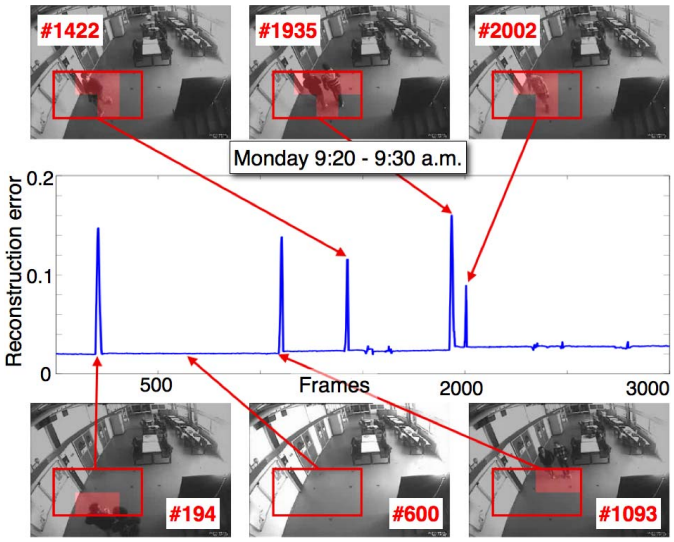


Fig. 8. A zoom in on the reconstruction error in a small time span on a region of interest marked with a red rectangle shows how the peaks correspond to interesting events only.

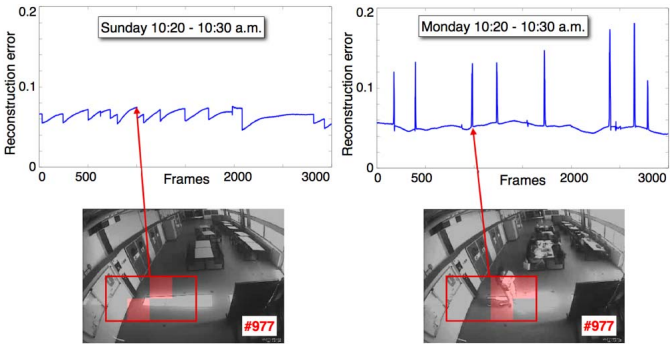


Fig. 9. A zoom in on the reconstruction error to compare the average reconstruction errors during sunday (left) and monday (right) morning. Here we notice how, on similar illumination condition, the monday morning trend is much more stable and the peaks correspond to actual dynamic events.

As a first thing, Fig. 6 shows the effect of changing the patch size and the amount of overlap. We consider all the videos of the *Baseline* set, one at a time, to analyze the possible relationships between the parameters and the size of the objects in the scene. On the left we show the effect of varying the amount of overlap once the patch size has been fixed to 40×40: in all the cases we notice that an increase of overlap improves the results. On the right we report the results obtained for different patch sizes and no overlap. In this case it is more difficult to choose the best size, even if we may infer a relationship between an appropriate size and the type of video (or else, the average size of the moving objects): for instance the *Highway* video contains small as well as big objects, thus a medium patch size seems to be advisable. In the *PETS2006* video the objects are quite small and distinctive w.r.t. the background, thus a small patch brings the highest performance, while in *Pedestrians* we find that a bigger patch brings better results, due to the fact that the people moving in the video, more or less always of the same size, project shadows that are more likely to be captured as foreground by small, thus more local, patches.

As a final cross check we may observe that a 40×40 patch with an overlap of 20 pixels (magenta bars on Fig. 6 - left) is better than a 20×20 patch with no overlap (violet bars on Fig. 6 - right). This seems to be due to the fact that, although the obtained resolution is equivalent, in the first case each 20×20 sub-patch is evaluated 4 times joint with different neighbors, leading to a redundant and more precise evaluation.

Table IV reports the results obtained by our method on the different categories, plus the overall results obtained by averaging the previous ones. The analysis shows the method performances according to different metrics: recall, precision, specificity, false positive rates (FPR), false negative rates (FNR), F-measure, percentage of bad classifications (PBC).

Table V compares our results with the most promising methods from the benchmark web site across all the categories. Since the statistics reported in the web site is rather rich, in this comparison we stick to the F-measure which summarizes the effect of false positives and false negatives. In these experiments the patch size was set to 40×40 or 32×32 pixels, depending on the video resolution,

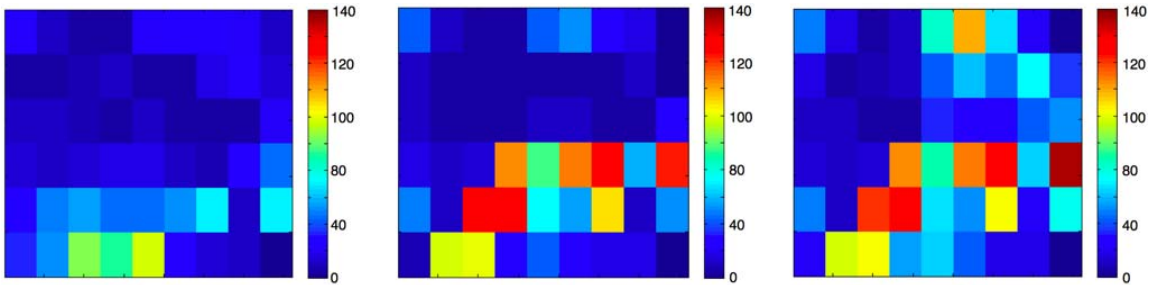


Fig. 10. Dictionary size after one hour acquisition on sunday (a), at the end of acquisition on sunday (b), at the of acquisition on monday (c).

and an overlap of half the patch size. Our overall result (F-measure = 0.77) is very close to the best performing algorithm (F-measure = 0.78), granting us the second position in the overall rank. Notice that the standard deviation associated with our performances is the lowest, and amounts to half the value of the best performing algorithm.

VI. Experiments Across Long Time Observations

In this section we analyze the performances of BMTDL across long time observations. We process a video stream covering two periods of time (each of 2 hours) of two consecutive days that have been selected to maximize their difference: Day 1 is a sunday morning (weekend) while Day 2 is a monday morning (working day). During the weekend we do not expect to observe any dynamic event caused by moving objects, but there will be smooth changes due to illumination. The monitored scenario is the one shown in Fig. 4.

Fig. 7 reports on two different plots the reconstruction errors obtained during the observation, averaged on all the patches. On sunday morning the trend of the reconstruction error was quite variable between 9:20 and 10:30 because of a blade of sun projected onto the floor (see Fig. 4), after that, the model remains quite stable. On monday morning we may observe spikes due to various dynamic events — people or groups coming into the hall.

Fig. 8 shows a detail of the reconstruction error trend on monday morning, between 9:20 and 9:30, on a meaningful region of interest (marked by the red rectangle). The red patches correspond to reconstruction errors above τ_{err}, set to 0.2 in these experiments. The height of the spikes is related to the area of the change, while its width to the temporal extent of the variation (within the whole region of interest).

Fig. 9, instead, compares the details of corresponding time frames acquired on sunday and monday morning. The sample frames show there is a similar illumination condition. On sunday the reconstruction error is fluctuating, since many dictionary updates (the value of τ_{temp} was fixed to $40''$) are required to accommodate new illumination conditions, while on monday we take advantage of the background model gathered the previous day and we obtain a very smooth reconstruction error with the exception of new dynamic events. Fig. 10 shows the per-patch dictionaries size reached at different times: on sunday morning, after one hour since the beginning of the experiments, most of the dictionaries are still very small ($K < 40$), with the exception of the lower part of the image plane where the blade of light is appearing. At the end of sunday morning, when the whole illumination variation event has been stored in the background model, the dictionaries of the lower part of the image plane are larger (10 patches have $100 \leq K \leq 180$), although the overall background model is still quite small (on average $K = 25.7$). On late monday the dictionaries size grew on the top right side of the image plane where people sit still for a long time, and end up being incorporated in the background. Dynamic changes occurring elsewhere are not incorporated in the background, as they should not be. By this time we have reached the maximum background size we may expect to obtain – from now on the sequence of update and cleaning steps will keep the overall size quite stable. In Fig. 11 we analyze the percentage of anomalies per patch: on sunday most of the anomalies occur on the bottom region, on monday on the top right regions. It is apparent that these regions will also trigger most of the dictionary updates.

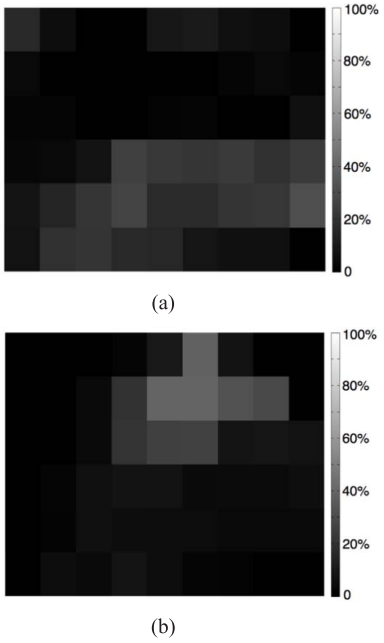


Fig. 11. Percentage of patches in an anomaly state on sunday (a) and monday (b).

TABLE VI
COMPUTATIONAL TIMES (FRAMES PER SECONDS) FOR DIFFERENT PATCH SIZES AND UPDATES FREQUENCY (SEE TEXT)

τ_{temp}	Patch size			
	10	20	40	80
200	0.77	1.52	4.01	3.74
800	0.34	2.9	9.30	21.10

We conclude with a remark on the dictionaries size. The current procedure for dictionary updating considers an injection of $K_{update} = 5$ atoms in order to incorporate most of the available up-to-date information. A spectral analysis on a sample of dictionaries allowed us to observe there is still some redundancy in our representation which lead us to the conclusion there is still some room for improving the compactness of our model. In particular, either the number of atoms inserted on each update, or the cleaning procedure, might be improved observing the significant singular values of the updated dictionaries, adapting the number of new atoms/the atoms to be removed to the effective information gained by the model. This issue is currently under investigation.

VII. CONCLUSION

We proposed a space-variant method for background modeling, we refer to as Background Modeling Through Dictionary Learning (BMTDL). The method is entirely data-driven, integrates an online dictionary learning procedure that allows us to update the model only when and where it is necessary, and is built on the assumption that, in the absence of dynamic events, all the video frames acquired by a fixed camera may be seen as a noisy version of a background prototype. On this respect the video stream produces semi-supervised input data which are fed to a dictionary learning module and produce a background model able to adapt to the working environment. The method, which is intrinsically patch based, leverages on the spatial correlation between pixels and shows robustness and stability across the different types of changes including illumination, jitter, motion, dynamic background, indoor/outdoor scenarios, and image sensors. A pixel based analysis of the obtained results reveals that the proposed module is already effective for detecting changes.

As a final remark, Table VI reports runtime complexities for 640×480 frames using an Intel 2.40GHz. We considered two different τ_{temp} (influencing the updates frequency and the training set size N during the updates) and four different patch sizes (controlling the data dimension n). The performances of our method strongly depend on both: if n is small the performances is mainly influenced by the number of updates, while if n is large the efficiency is affected by the cost of a single update, which becomes larger as N grows.

Notice that the proposed framework can be naturally parallelized on multicore systems, thanks to the inherent independence of the computation of each patch. The design of an appropriate architecture where dictionary update is (possibly) performed offline and the online processing of each patch is synchronized is currently under investigation.

REFERENCES

[1] R. G. Abbott and L. R. Williams, "Multiple target tracking with lazy background subtraction and connected components analysis," *Mach. Vis. Appl.*, vol. 20, no. 2, pp. 93–101, Feb. 2009.

[2] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[3] S. Atev, O. Masoud, and N. Papanikolopoulos, "Learning traffic patterns at intersections by spectral clustering of motion trajectories," in *Proc. IROS*, Oct. 2006, pp. 4851–4856.

[4] O. Barnich and M. Van Droogenbroeck, "ViBe: A universal background subtraction algorithm for video sequences," *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1709–1724, Jun. 2011.

[5] C. Basso, M. Santoro, A. Verri, and S. Villa, "PADDLE: Proximal algorithm for dual dictionaries LEarning," in *Proc. 21st Int. Conf. ICANN*, 2011, pp. 379–386.

[6] T. Bouwmans, "Recent advanced statistical background modeling for foreground detection: A systematic survey," *Recent Patents Comput. Sci.*, vol. 4, no. 3, pp. 147–176, Sep. 2011.

[7] S. Brutzer, B. Hoferlin, and G. Heidemann, "Evaluation of background subtraction techniques for video surveillance," in *Proc. IEEE Conf. CVPR*, Jun. 2011, pp. 1937–1944.

[8] V. Cevher, A. Sankaranarayanan, M. F. Duarte, D. Reddy, R. G. Baraniuk, and R. Chellappa, "Compressive sensing for background subtraction," in *Computer Vision* (Lecture Notes in Computer Science), vol. 5303. Berlin, Germany: Springer-Verlag, 2008, pp. 155–168.

[9] R. Chang, T. Gandhi, and M. M. Trivedi, "Vision modules for a multisensory bridge monitoring approach," in *Proc. 7th Int. IEEE Conf. Intell. Transp. Syst.*, Oct. 2004, pp. 971–976.

[10] S. Cohen, "Background estimation as a labeling problem," in *Proc. 10th IEEE ICCV*, vol. 2. Oct. 2005, pp. 1034–1041.

[11] R. Cucchiara, "Multimedia surveillance systems," in *Proc. 3rd ACM Int. Workshop Video Surveill. Sensor Netw.*, 2005, pp. 3–10.

[12] C. David and V. Gui, "Sparse coding and Gaussian modeling of coefficients average for background subtraction," in *Proc. 8th ISPA*, Sep. 2013, pp. 230–235.

[13] C. Demant, B. Streicher-Abel, P. Waszkewitz, M. Strick, and G. Schmidt, *Industrial Image Processing: Visual Quality Control in Manufacturing*. New York, NY, USA: Springer-Verlag, 1999.

[14] M. Dikmen, S.-F. Tsai, and T. S. Huang, "Base selection in estimating sparse foreground in video," in *Proc. 16th IEEE ICIP*, Nov. 2009, pp. 3217–3220.

[15] M. Dikmen and T. S. Huang, "Robust estimation of foreground in surveillance videos by sparse error estimation," in *Proc. ICPR*, Dec. 2008, pp. 1–4.

[16] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *Proc. ECCV*, 2000, pp. 751–767.

[17] A. Fabijanska, M. Kuzanski, D. Sankowski, and L. Jackowska-Strumitto, "Application of image processing and analysis in selected industrial computer vision systems," in *Proc. Int. Conf. Perspect. Technol. Methods MEMS Design*, May 2008, pp. 27–31.

[18] M. Fornasier, A. Langer, and C.-B. Schönlieb, "A convergent overlapping domain decomposition method for total variation minimization," *Numer. Math.*, vol. 116, no. 4, pp. 645–685, Oct. 2010.

[19] I. Gori, S. R. Fanello, F. Odone, and G. Metta, "All gestures you can: A memory game against a humanoid robot," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, Nov./Dec. 2012, pp. 330–336.

[20] C. Guyon, T. Bouwmans, and E.-H. Zahzah, "Foreground detection based on low-rank and block-sparse matrix decomposition," in *Proc. 19th IEEE ICIP*, Sep./Oct. 2012, pp. 1225–1228.

[21] W. Hu, D. Xie, Z. Fu, W. Zeng, and S. Maybank, "Semantic-based surveillance video retrieval," *IEEE Trans. Image Process.*, vol. 16, no. 4, pp. 1168–1181, Apr. 2007.

[22] X. Huang, F. Wu, and P. Huang, "Moving-object detection based on sparse representation and dictionary learning," in *Proc. Conf. Comput. Intell. Bioinformat.*, vol. 1. 2012, pp. 492–497.

[23] Z. Ji, W. Wang, and K. Lu, "Extract foreground objects based on sparse model of spatiotemporal spectrum," in *Proc. 20th IEEE ICIP*, Sep. 2013, pp. 3441–3445.

[24] N. Johnson and D. Hogg, "Learning the distribution of object trajectories for event recognition," *Image Vis. Comput.*, vol. 14, no. 8, pp. 609–615, Aug. 1996.

[25] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time

foreground-background segmentation using codebook model," *Real-Time Imag.*, vol. 11, no. 3, pp. 172–185, Jun. 2005.

[26] S. Kong and D. Wang, "A dictionary learning approach for classification: Separating the particularity and the commonality," in *Proc. ECCV*, 2012, pp. 186–199.

[27] A. Langer, M. Fornasier, and C.-B. Schönlieb, "Domain decomposition methods for compressed sensing," in *Proc. Int. Conf. Sampling Theory Appl.*, 2009.

[28] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Proc. NIPS*, 2006, pp. 801–808.

[29] H.-Y. M. Liao, D.-Y. Chen, C.-W. Sua, and H.-R. Tyan, "Real-time event detection and its application to surveillance systems," in *Proc. IEEE ISCAS*, May 2006, pp. 509–512.

[30] H.-H. Lin, T.-L. Liu, and J.-H. Chuang, "Learning a scene background model via classification," *IEEE Trans. Signal Process.*, vol. 57, no. 5, pp. 1641–1654, May 2009.

[31] C. Lu, J. Shi, and J. Jia, "Online robust dictionary learning," in *Proc. IEEE Conf. CVPR*, Jun. 2013, pp. 415–422.

[32] V. Mahadevan and N. Vasconcelos, "Spatiotemporal saliency in dynamic scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 171–177, Jan. 2010.

[33] D. Makris and T. Ellis, "Path detection in video surveillance," *Image Vis. Comput.*, vol. 20, no. 12, pp. 895–903, Oct. 2002.

[34] S. Mallat, *A Wavelet Tour of Signal Processing*, 2nd ed. San Francisco, CA, USA: Academic, 1999.

[35] S. Messelodi, C. M. Modena, N. Segata, and M. Zanin, "A Kalman filter based background updating algorithm robust to sharp illumination changes," in *Proc. ICIAP*, Sep. 2005, pp. 163–170.

[36] B. T. Morris and M. M. Trivedi, "Trajectory learning for activity understanding: Unsupervised, multilevel, and long-term adaptive approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 11, pp. 2287–2301, Nov. 2011.

[37] N. Noceti, A. Destrero, A. Lovato, and F. Odone, "Combined motion and appearance models for robust object tracking in real-time," in *Proc. AVSS*, Sep. 2009, pp. 412–417.

[38] N. Noceti and F. Odone, "Learning common behaviors from large sets of unlabeled temporal series," *Image Vis. Comput.*, vol. 30, no. 11, pp. 875–895, Nov. 2012.

[39] B. Packer, K. Saenko, and D. Koller, "A combined pose, object, and feature model for action understanding," in *Proc. IEEE Conf. CVPR*, Jun. 2012, pp. 1378–1385.

[40] K. A. Patwardhan, G. Sapiro, and V. Morellas, "Robust foreground detection in video using pixel layers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 4, pp. 746–751, Apr. 2008.

[41] P. W. Power and J. A. Schoonees, "Understanding background mixture models for foreground segmentation," in *Proc. Image Vis. Comput.*, Nov. 2002, pp. 267–271.

[42] A. Mittal and N. Paragios, "Motion-based background subtraction using adaptive kernel density estimation," in *Proc. CVPR*, vol. 2. Jun./Jul. 2004, pp. II-302–II-309.

[43] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam, "Image change detection algorithms: A systematic survey," *IEEE Trans. Image Process.*, vol. 14, no. 3, pp. 294–307, Mar. 2005.

[44] M. Ranzato, F. J. Huang, Y.-L. Boureau, and Y. LeCun, "Unsupervised learning of invariant feature hierarchies with applications to object recognition," in *Proc. IEEE Conf. CVPR*, Jun. 2007, pp. 1–8.

[45] O. Schreer, I. Feldmann, U. Golz, and P. Kauff, "Fast and robust shadow detection in videoconference applications," in *Proc. Int. Symp. Video/Image Process. Multimedia Commun.*, 2002, pp. 371–375.

[46] M. Seki, T. Wada, H. Fujiwara, and K. Sumi, "Background subtraction based on cooccurrence of image variations," in *Proc. IEEE Comput. Soc. Conf. CVPR*, vol. 2. Jun. 2003, pp. II-65–II-72.

[47] R. Sivalingam, A. D'Souza, M. Bazakos, R. Miezianko, V. Morellas, and N. Papanikolopoulos, "Dictionary learning for robust background modeling," in *Proc. IEEE ICRA*, May 2011, pp. 4234–4239.

[48] A. Staglianò, G. Chiusano, C. Basso, and M. Santoro, "Learning adaptive and sparse representations of medical images," in *Proc. Med. Comput. Vis. Recognit. Techn. Appl. Med. Imag.*, vol. 6533. 2011, pp. 130–140.

[49] A. Staglianò, N. Noceti, F. Odone, and A. Verri, "Background modeling through dictionary learning," in *Proc. 20th IEEE ICIP*, Sep. 2013, pp. 2524–2528.

[50] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Comput. Soc. Conf. CVPR*, vol. 2. Jun. 1999, pp. 246–252.

[51] C. Stauffer and K. Tieu, "Automated multi-camera planar tracking correspondence modeling," in *Proc. IEEE Comput. Soc. Conf. CVPR*, vol. 1. Jun. 2003, pp. I-259–I-266.

[52] M. Taj and A. Cavallaro, "Distributed and decentralized multicamera tracking," *IEEE Signal Process. Mag.*, vol. 28, no. 3, pp. 46–58, May 2011.

[53] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Statist. Soc. B*, vol. 58, no. 1, pp. 267–288, 1996.

[54] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and practice of background maintenance," in *Proc. ICCV*, vol. 1. Sep. 1999, pp. 255–261.

[55] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proc. IEEE Conf. CVPR*, Jun. 2010, pp. 3360–3367.

[56] D. Weinland, R. Ronfard, and E. Boyer, "Free viewpoint action recognition using motion history volumes," *Comput. Vis. Image Understand.*, vol. 104, nos. 2–3, pp. 249–257, Nov./Dec. 2006.

[57] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 780–785, Jul. 1997.

[58] D. Wu, F. Zhu, and L. Shao, "One shot learning gesture recognition from RGBD images," in *Proc. IEEE Comput. Soc. Conf. CVPRW*, Jun. 2012, pp. 7–12.

[59] M. Wu and X. Peng, "Spatio-temporal context for codebook-based dynamic background subtraction," *AEU—Int. J. Electron. Commun.*, vol. 64, no. 8, pp. 739–747, Aug. 2010.

[60] W. Xian, Y. Zhang, Z. Tu, and E. L. Hall, "Automatic visual inspection of the surface appearance defects of bearing roller," in *Proc. IEEE ICRA*, vol. 3. May 1990, pp. 1490–1494.

[61] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. IEEE Conf. CVPR*, Jun. 2009, pp. 1794–1801.

[62] Y. Zhang and Q. Ji, "Active and dynamic information fusion for facial expression understanding from image sequences," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 699–714, May 2005.

[63] C. Zhao, X. Wang, and W.-K. Cham, "Background subtraction via robust dictionary learning," *J. Image Video Process.*, Jan. 2011.

Alessandra Staglianò received the Laurea degree in computer science and the Ph.D. degree from the University of Genova, in 2009 and 2014, respectively. In 2011, she visited the University College of London. Her research activities include machine learning and its application to computer vision problems. She is particularly interested into the extraction of meaningful information from big data, with application to medical imaging, image and video analysis, and action and emotion recognition. Since 2013, she has been involved in the EU FP7-ICT ASC-Inclusion Project.

Nicoletta Noceti received the Laurea (*cum laude*) and the Ph.D. degrees in computer science from the University of Genova, in 2006 and 2010, respectively. In 2008, she visited the IDIAP Institute, Switzerland. Since 2010, she has been a Research Associate (Post-Doc) at DIBRIS, University of Genova.

Her research interests include computer vision and machine learning, with application to image and video analysis, activity and action classification, object detection and recognition, and scene understanding.

She has participated to various national and international research projects, and to technology transfer and development projects with PMI and big companies. She collaborates with various research institutes, such as Idiap Research Institute, IIT, and the Massachusetts Institute of Technology.

Alessandro Verri received the Laurea and Ph.D. degrees in physics from the University of Genova, in 1984 and 1988, respectively. Since 2000, he has been a Professor of Computer Science with the University of Genova. He was a Visiting Scientist and Professor with the Massachusetts Institute of Technology several times from 1986 to 2013. His scientific interests include learning theory, learning algorithms, biomedical and molecular biology data analysis, and computer vision. He coordinated and coordinates many projects of basic and applied research and technology transfer. He has authored over 120 papers and co-authored a book on *computer vision* with E. Trucco published by Prentice Hall.

Francesca Odone received the Laurea (*cum laude*) degree in information sciences and the Ph.D. degree in computer science from the University of Genova, in 1997 and 2002, respectively. She visited Heriot-Watt University, Edinburgh, U.K., in 1997, as a Research Associate, and was a Visiting Ph.D. student with an EU Marie Curie Research Grant, in 1999. From 2002 to 2005, she was a Researcher at the Institute for the Physics of Matter, National Research Center. Since 2014, she has been an Associate Professor with the Department of Informatics, Bioengineering, Robotics, and Systems Engineering, University of Genova. She has authored about 70 papers on international journals and conference proceedings covering many aspects of computer vision and machine learning. Her research interests include learning representations for high dimensional data: (structured) feature selection, dimensionality reduction, support set estimation, visual recognition pipelines for object detection, retrieval, and recognition in images and image sequences, and algorithms for behavior understanding and action recognition.