Nobody Likes Mondays: Foreground Detection and Behavioral Patterns Analysis in Complex Urban Scenes

Gloria Zen*
DISI, University of Trento
Trento, Italy
zen@disi.unitn.it

John Krumm
Microsoft Research
Redmond, Washington
jckrumm@microsoft.com

Nicu Sebe
DISI, University of Trento
Trento, Italy
sebe@disi.unitn.it

Eric Horvitz
Microsoft Research
Redmond, Washington
horvitz@microsoft.com

Ashish Kapoor
Microsoft Research
Redmond, Washington
akapoor@microsoft.com

ABSTRACT

Streams of images from large numbers of surveillance webcams are available via the web. The continuous monitoring of activities at different locations provides a great opportunity for research on the use of vision systems for detecting actors, objects, and events, and for understanding patterns of activity and anomaly in real-world settings. In this work we show how images available on the web from surveillance webcams can be used as sensors in urban scenarios for monitoring and interpreting states of interest such as traffic intensity. We highlight the power of the cyclical aspect of the lives of people and of cities. We extract from long-term streams of images typical patterns of behavior and anomalous events and situations, based on considerations of day of the week and time of day. The analysis of typia and atypia required a robust method for background subtraction. For this purpose, we present a method based on sparse coding which outperforms state-of-the-art works on complex and crowded scenes.

Categories and Subject Descriptors

I.4 [**Image Processing and Computer Vision**]: Miscellaneous; I.2.10 [**Vision and Scene Understanding**]: [Video Analysis]

Keywords

video surveillance; long-term pattern analysis; background subtraction; unsupervised feature learning; auto-encoders;

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
ARTEMIS '13 Barcelona, Spain
Copyright 2013 ACM 978-1-4503-2393-2/13/10 ...$15.00.
http://dx.doi.org/10.1145/2510650.2510653.

*This work was performed during an internship at Microsoft Research.

1. INTRODUCTION

Webcams provide sets of eyes on the world, continuously collecting data on scenes ranging from natural phenomena to human activities. Numerous streams of webcam data are publicly available, providing opportunities for studies of patterns of events over extended periods of time. Analysis of events over time on a view of a region of a city could provide useful data for guiding the decongestion of traffic, supporting urban planning, and understanding human behavior. However, with the exception of a few works [1, 4, 14], little effort has been focused on harnessing such imagery. Directions of research in this realm include the identification of typical patterns of behavior over a day or month, and based on such regular patterns, identifying anomalous events and situations.

In this work, we analyze a full month of data acquired for a complex urban scene - a specific view on Fifth Avenue in New York City. We show how meaningful patterns of activities can be extracted from the streams of images. Previous studies on long-term visual surveillance have focused on the analysis of the video sequence considering the entire frame as a whole [1, 4, 14]. In contrast, we aim at performing separation of the *foreground* (FG) from the *background* (BG), in order to (i) eliminate the high data redundancy present in the BG and (ii) better focus on the behavior of the foreground elements of interest, which in our scenario correspond to cars and pedestrians.

Our proposal employs the most promising method reported in the literature [3, 5] and build on this approach to achieve better performance. We found that the ability of the best reported method does not reliably identify foreground in the real-world urban imagery at the focus of our study. We believe that our dataset is particularly difficult w.r.t. the task of FG detection because of the following challenges: (i) high *complexity* of the scenario, *i.e.* high variability of the BG, which is subject to strong appearance changes because of light and weather variation; (ii) *crowdedness* of the scene, with many targets (*i.e.* cars and pedestrians) present especially during rush hours; (iii) low frame-rate acquisition, which makes the problem of sudden illumination changes more crucial; (iv) low signal-to-noise (S/N) ratio at night; (v) high incidence of light reflections, especially at night and during rain, which causes FG false detection; (vi) presence of

Figure 1: NYC-5th dataset: representative frames of a *complex* and *crowded* scene, challenging for a foreground detection task based on background modeling.

jpeg artifacts in streamed webcam imagery. Sample frames illustrating some of these issues are shown in Fig.1.

In order to increase the reliability of the foreground extraction, we introduce a new method which builds upon [32] but significantly improves the performance by relying on more informative (sparse) features beyond *rgb* information. Additionally, we show that a simple and efficient measure based on FG detection is strongly correlated with high level information, thus revealing very interesting insights about typical patterns of a city life, as well as anomalies.

2. RELATED WORK

Some of the key challenges associated with the analysis of data extracted from video over extended periods include requirements for storage and long processing times. To reduce storage and make processing more efficient, studies of long-term video surveillance typically rely on video collected at low frame rates [4, 14]. However, low frame rates can make it difficult to reconstruct motion tracks for analyzing behavioral patterns. Another way to deal with the scale is to only collect data streams that are essentially static, *i.e.* the background appearance varies in time due to light, weather or seasonal changes but almost no foreground elements can be observed [14].

While there exists prior research on detecting anomalies in urban scenarios [4], the previous works focus on video analysis at frame level (*e.g.* extracting a pyramid of feature histograms). Further, there are hardly any approaches that attempt to distinguish the foreground elements from the background and to analyze their behavior separately. One of the exception is Abrams *et al.* [1] recorded a data set (LOST) with high frame rate from 17 cameras for over one year, in order to explore the changes in daily tracks. They record the same half hour each day, limiting the long-term analysis to a short interval in each day, and show that histograms of track density have a high-level interpretation w.r.t. natural human behavior. We show that this analysis can be done in a more efficient and at a higher time granularity way by exploiting the FG signal.

One of the core component of our work is a background subtraction module. In visual surveillance with static cameras, a BG subtraction method based on BG modeling is typically adopted. We considered the set of techniques described in recent surveys [3, 5] to select the most promising approach for our goals. The most widely used method was proposed by Stauffer and Grimson [24]. Among variations of this approach, [32] appears to be the most robust to the dynamics we face with the analysis of long-term streams, including *dynamic background*, *darkening*, and *noise during night time*. However, these techniques were not very successful on our long-term sequences (see Fig.4, 5 and supplementary material[1]). The results were fairly poor especially for the case of sudden light changes, a problem which is accentuated by the low frame acquisition rate. Also these approaches performed poorly during night because of low signal-to-noise ratio and presence of light reflections.

Our hypothesis in this work is that BG subtraction methods relying only on pixel-based analyses are not powerful enough, and leveraging more informative features (*e.g.* based on local structure) may be valuable. The use of local features like HOG [10] or texture has been proposed [13, 19, 27]. To our knowledge, no work has yet explored the potential of using sparse features extracted at *patch*-level for the background subtraction task. We show the effectiveness of using features learned via auto-encoders [17]. Such methods for learning features from data in an unsupervised manner have been applied successfully in a variety of fields (NLP, audio, computer vision, etc.), Most of the research in computer vision on using learned versus hand-designed features has focused on classification tasks like object recognition [2, 23, 29], image classification, [18, 22] or facial expression recognition [21], where the unsupervised phase of feature learning is combined with a supervised training phase of a classifier.

In summary, we have observed that prior methods for background subtraction are not adequate to our problem scenario because: (i) our dataset has a very low frame rate precluding the use of temporal information as is done in [31], (ii) many of the FG detection methods rely on the assumption that the FG information is sparse [6, 7, 8, 9, 11, 30]. The latter assumption is not valid in our case where we routinely encounter crowded scenes (see Fig.1).

3. OUR METHOD

Our method employs two main steps: (i) FG detection based on BG modeling and, (ii) analysis of behavioral patterns, based on the extracted FG information (Sec.3.3). In order to improve the performance at step (i), we propose a new BG subtraction method where the BG model is built on a sparse representation via a feature dictionary that is learned from the input data (Sec.3.2). We provide a general overview on Sparse Coding and Auto-Encoders in Sec.3.1.

3.1 Learning Local Features Dictionary with Auto-Encoders

Sparse Coding. In sparse signal modeling, input signals are represented as a (often linear) combination of a few coefficients selecting atoms in some over-complete bases or dictionary $D = \{\Phi_j\}$. Formally,

$$x = \sum_{j=1}^{M} a_j \Phi_j + \epsilon \quad (1)$$

[1]http://disi.unitn.it/~zen/video/artemis13.avi

Here $x \in \mathbb{R}^N$, $\mathbf{a} = \{a_j\} \in \mathbb{R}^M$, $D = \{\Phi_j\} \in \mathbb{R}^{N \times M}$, generally $M > N$, and ϵ is the approximation error. Note that when $\mathbf{a}$ is sparse, most of the bins of $\mathbf{a} \in \mathbb{R}^M$ are zero. Formally, we can write this sparsity condition as $||a_j||_0 = K$, $K < M$. In other words, while x is mapped to a higher dimensional space (*i.e.* from $\mathbb{R}^N$ to $\mathbb{R}^M$, with $M > N$), generally the dimension of its sparse representation is lower than the initial space dimension (*i.e.* $K < N < M$).

It has been shown that mapping the data into a significantly higher dimensional space with an over-complete basis dictionary can lead to superior performance in many applications [18, 26]. In this work, we investigate the effect of using sparse coding for background modeling. Relative to prefixed dictionaries such as wavelets, learned dictionaries bring the advantage of better adapting to the images, thereby enhancing the sparsity [28]. We learn our basis dictionary $D = \{\Phi_j\} \in \mathbb{R}^{N \times M}$ through sparse Auto-Encoders. We briefly review auto-encoders below. For more detailed explanation we refer readers to [17].

Auto-Encoders. Auto-encoders (AE) are unsupervised models that learn a compressed representation for a set of data. Specifically, auto-encoders learn a function $h(\cdot)$ that maps an input vector $x \in \mathbb{R}^N$ to a feature vector $a = h(x) \in \mathbb{R}^M$, together with a function $g(\cdot)$, that maps $h(x)$ back to $\hat{x} = g(h(x)) \in \mathbb{R}^N$, where $\hat{x}$ is the reconstruction of the input vector x. In the AE notation, N and M are respectively the number of *visible* and *hidden* units.

The functions $h(\cdot)$ and $g(\cdot)$, named respectively *encoder* and *decoder* function, are computed in a way that minimizes the reconstruction error between the two vectors x and $\hat{x}$. The Encoder function $h(\cdot)$ is defined as: $a = h_{W,b}(x) = s(W_1 x + b) = s(z)$, where s is the *activation function*. In this case we chose s to be the sigmoid function $s(z) = \frac{1}{1+e^{-z}}$. Estimating $h(\cdot)$ corresponds to the estimation of the parameters $W \in \mathbb{R}^{M \times N}$, which is the *weight matrix*, and $b \in \mathbb{R}^M$, which is the *bias vector*. The Decoder function $g(\cdot)$ is defined as: $g_{W,c}(z) = s(W_2 z + c)$. Given a set of p input vectors $x^{(i)}, i = 1, ..., p$, the weight matrices W_1 and W_2 are adapted using backpropagation to minimize the reconstruction error. The cost function to be minimized is therefore:

$$J(W, b) = \min_{W,b} \sum_{i=1}^{p} ||x^{(i)} - \hat{x}^{(i)}||_2 \tag{2}$$

where $\hat{x}^{(i)}$ is dependent implicitly on $\{W, b\}$ and $||\cdot||_2$ is the Euclidean distance. This step can be performed via batch gradient descent or more sophisticated algorithms like conjugate gradient or L-BFGS to speed up the performance [17]. A penalty term is also added in the optimization function to force the learned features to have desirable properties. There are many sophisticated versions of auto-encoders; differences arise essentially in the specifics of the assigned penalty term. In our case, we use sparse auto-encoders, which force the average hidden unit activation to be sparse [12]. This is done by designing a *penalty term*, which enforces the activation of a hidden unit $\hat{\rho}_j = \frac{1}{p} \sum_{i=1}^{p} [a_j(x^{(i)})]$, $j = 1, \cdots, M$ to be close to a desired value, $\hat{\rho}_j = \rho$, where ρ is the *sparsity parameter*. The overall cost function is now:

$$J_{sparse}(W, b) = J(W, b) + \beta \sum_{j=1}^{M} KL(\rho || \hat{\rho}_j) \tag{3}$$

where the weight parameter β controls the relative importance of the *sparsity penalty* term, and KL is the Kullback Leibler distance.

A *weight decay* term on W is also added, whose importance is regulated by the parameter λ, in order to penalize the magnitude of the weight and prevent overfitting. The parameters used in this framework are therefore: (i) the number of *hidden units* M; (ii) the *sparsity* parameter ρ; (iii) the *weight* of the *sparsity penalty* term β; (iv) the *weight decay* parameter λ.

Learning Feature Dictionary for Sparse Representation of Local Patches. In the context of generic image understanding, instead of the whole image sparse coding is applied to local parts or descriptors [26]. In these approaches, the input signal x usually corresponds to a small image patch of $\sqrt{N} \times \sqrt{N}$ pixels which is stacked as a vector $x \in \mathbb{R}^N$. Similarly our approach first randomly samples a set of p patches $\{x_i\}$ from a sufficiently representative training sequence of images. Auto-encoders with parameters $\theta = [W, b]$ are then learnt using these samples. The feature dictionary discovered thus captures the most basic and typical visual patterns presented in the training images set.

3.2 Background Subtraction

We build upon [32] for background modeling. In our work besides the *rgb* values, we propose to include more discriminative auto-encoder features in the background model. For each pixel *(i,j)* we consider the sparse representation $a \in \mathbb{R}^M$, obtained by mapping the patch centered at *(i,j)* via the *Encoder* function $h(\cdot)$ (previously learned, as explained in Section 3.1). Note that general sparse methods for background subtraction consider the image as a whole [6, 8, 9, 11] and no attempt has been made to date to build a sparse vocabulary of local patches. The background model for each pixel is then built as in [32], modeling the pixel features distribution as a Mixture of K Gaussians:

$$p_i(x) = \sum_{k=1}^{K} \pi_k e^{-\frac{1}{2}(x-\mu_k)' \Sigma_k^{-1} (x-\mu_k)} \tag{4}$$

where $\Sigma \in \mathbb{R}^{D \times D}$ and $D = 3 + M$. For computational efficiency, an assumption of dimensionality independence is made. In this way, the full covariance matrix simplifies to a diagonal matrix: $diag(\Sigma) = [\sigma_1^2, ..., \sigma_D^2]$. A second assumption that the variance is the same in each direction (*i.e.* $\sigma_1 = \sigma_2 = ... = \sigma_D$) simplifies Σ as follows: $\Sigma = \sigma^2 I$, where $I \in \mathbb{R}^{D \times D}$ is the identity matrix. This simplifies formula 4 as follows:

$$p_i(x) = \sum_{k=1}^{K} \pi_k e^{-\frac{(x-\mu_k)'(x-\mu_k)}{2\sigma_k^2}}$$

While these assumptions have proven to be effective for a mixture of Gaussians based on *rgb* features, a priori, we cannot expect that methods based on them will work with different types of features. We will see in Section 4 that these assumptions also hold in our case. The parameters involved in this framework of adaptive mixture model for BG subtraction are the learning rate α and the threshold T_b, that decides if a point data is well described by the BG model or not [32].

3.3 Extracting Typical and Anomalous Patterns of Behavior

Previous works [15, 16, 20] showed that statistical analyses based on simple low-level cues, *e.g.* optical flow, can reveal high-level recurrent patterns of behaviors. However, these analyses are performed on short-term periods, *e.g.* several hours of video. Thus, the recurrent behaviors extracted correspond to such examples as different traffic flows regulated by traffic lights. In this work, we wish to extract significant patterns correlated to human behaviors which are exhibited via long-term analyses; we show how such analyses can be done via a simple measure such as the percentage of foreground pixels, here denoted by τ. The intuition here is that the high-level information τ can be interpreted as an intensity measure of activities happening in a region of interest. For example, when few vehicles are circulating, *e.g.* in the early morning, τ will be low, while during rush hours the measured τ will be higher. An anomalous behavior can be determined by considering the *agreement* on τ of observations taken at a specific day of the week and at a specific time of the day. In details, given N observations τ_i, *with* $i = 1, \ldots, N$, we define the anomaly score as the variance-scaled distance from τ_i to μ_τ:

$$S_i = \frac{|\tau_i - \mu_\tau|}{\sigma_\tau} \tag{5}$$

Intuitively, given N observations taken *e.g.* on Monday $9am$, the anomaly score for τ_1 will be higher, w.r.t. $S_2, \cdots, S_N$, if its agreement on μ_τ is lower, w.r.t. agreement given by $\tau_2, \cdots \tau_N$, and so on. Experimental results are in line with our assumptions and they will be discussed in Section4.

4. EXPERIMENTAL RESULTS

In this section, we present the dataset and the results obtained with our method.

4.1 Dataset

Our dataset consists of imagery collected from a view of Fifth Avenue of New York City (NYC-5th). The stream was collected from a webcam that is part of the EarthCam Network[2]. The data was collected over nearly four weeks, from December 1^{st} to 25^{th}, 2011, at a rate of 2 frames per minute. Frame size is 480×640 pixels. The collected $\sim 72K$ frames require a $12Gb$ storage occupancy. Acquiring the same temporal period at a rate of 1 *fps* would have required 225 *Gb* storage. Note that the dataset was acquired with +5 hours shift w.r.t. local time. Thus the imagery data available from $0{:}00am$ to $11{:}59pm$ per each day corresponds, to the local NYC time, per the time lapse between $7{:}00pm$ and $6{:}59pm$. We are going to present the daily patterns of behavior within the latter time interval. In order to evaluate the accuracy of FG detection, on which we rely for further analysis, a sequence of frames has been annotated with the ground-truth FG mask. In particular, two frames per hour, on Dec.6^{th}, have been annotated (*i.e.* we picked frames at times 00:00, 00:30, 01:00, and so on). We selected this day as it contains a large variety of light and weather changes. Data and ground truth are available online on the author's website[3]. Our hope is that this dataset can contribute as a reference benchmark for long-term activity analysis, as well as for background subtraction (BS) methods evaluation on complex and crowded sequences. Available benchmarks for BS evaluation, generally consist of short sequences, with a few annotated frames [25], or consist of artificially generated sequences [5].

[2]http://www.earthcam.com/usa/newyork/fifthave/?cam=nyc5th_str

[3]http://disi.unitn.it/~zen

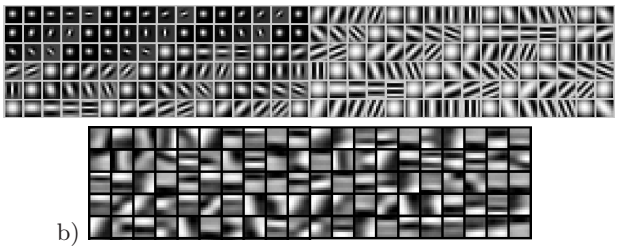

Figure 2: Examples of (a) Gabor Filters and (b) features learned with Auto-Encoders.

4.2 Learning a Vocabulary for Sparse Patch Representation

As a first step, we train the auto-encoders in order to generate the features that allow a sparse representation of the video data. We set the dimension of the patch to $\sqrt{N} = 8$. This value is a good compromise as it allows us to learn sufficiently discriminative features; larger patches would have a higher probability of including foreground. Figure 2(a) shows the set of features learned by randomly sampling in space and time $p = 20000$ patches from a 1-day-long sequence (*i.e.* Dec.6^{th}) and using the following setting: $M = 128$, $\rho = 0.003$, $\beta = 3$, and $\lambda = 0.0001$. Training the auto-encoders with these settings on an Intel(R) Core(TM) i5 CPU, 2.67GHz requires about 20 minutes.

Figure 2(a) can help with understanding the meaning of the weight matrix W and the hidden representation a. Each column of the weight matrix $W \in \mathbb{R}^{N \times M}$ is reshaped to form a $\sqrt{N} \times \sqrt{N}$ patch. The $M = 128$ filters obtained are displayed. When a patch x is mapped via $h(\cdot)$ to a $\mathbf{a} = \{a_j\} \in \mathbb{R}^M$ sparse representation, the non-zero coefficients of $\mathbf{a}$ identify the features that better represent the signal x. It is well known that features learned via auto-encoders at one layer resemble Gabor-like filters. However, it is also known that learning features from data often achieves a sparser representation as the learned dictionary better fits the data [28]. For example, looking at Fig.2(a) w.r.t. Fig.2(b), we can see that filters with a certain diagonal orientation (from top left to bottom right) are nearly absent because the scene perspective is one where a majority of edges are oriented on the other diagonal direction. Additionally, beyond regularly oriented edge gradients, some data-specific shapes can be learned. *Learning* w.r.t. *using hand-designed* features may be a promising approach in the context of visual surveillance and in crowded scenarios like the one we considered, in which: i) both BG and FG signal present a high data redundancy, where the background is constant and similar elements tend to appear repetitively in space and time, and ii) the variability of features data is limited to the *small world* variability defined by the scene observed through the camera.

4.3 Foreground Detection

The performance evaluation of the different methods for FG detection is reported in Fig.3 and Tab.1. The performance is measured in term of *precision* and *recall* which

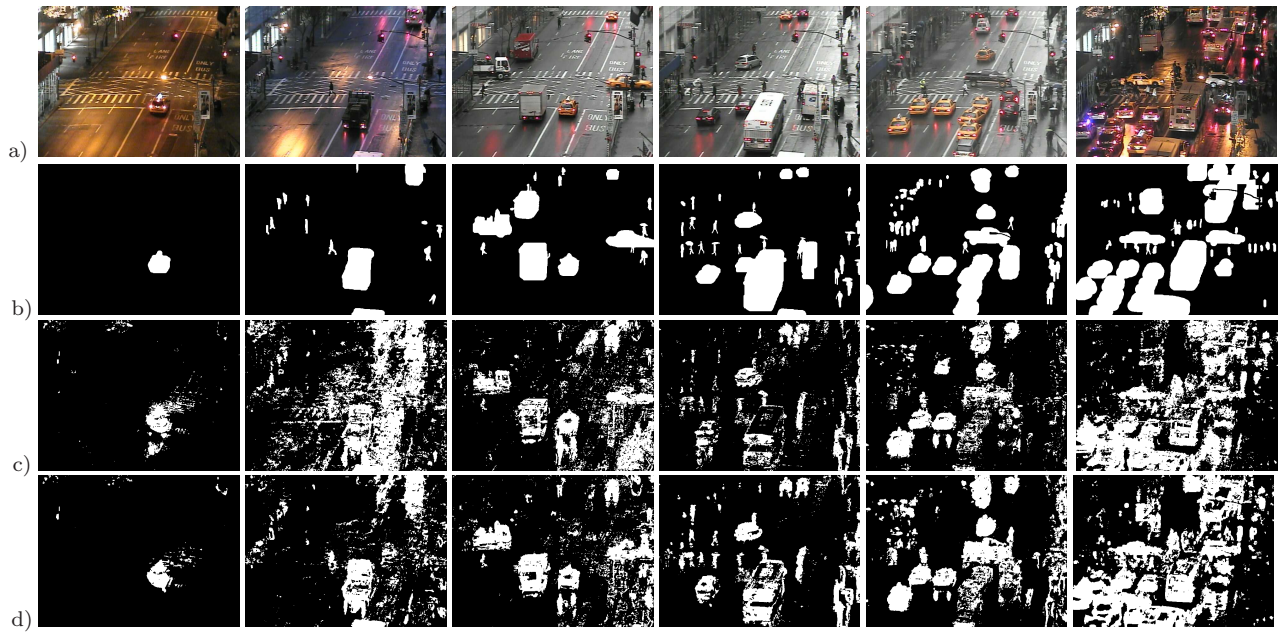

Figure 4: Performance on foreground detection. (a) original frame (b) ground truth (c) [32] (d) our method.

quantify, respectively, the number of pixels correctly identified as FG, divided by the number of pixels classified as FG (p) and by the number of pixels defined as FG in the ground truth (r). The F-measure is defined as: $F = (1+\beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}$, with $\beta = 1$. Fig.3(a) shows the performance of the three methods at varying parameters T_b and α (in details, at varying $T_b^{rgb}, T_b^{ae} \in [3.5, 4.5]$, $T_b^{rgb-ae} \in [7.0, 8.0]$, and $\alpha \in [0.01, 0.10]$).

Also, the average error on the FG percentage estimation, ϵ_τ, has been measured. It was observed that the best estimate of τ is obtained with the highest balance between precision and recall (this optimal results area is highlighted in red in Fig.3(a)). Among all possible (T_b, α) values combinations, the one allowing the best performance for each method is shown in Tab.1.

In general, it can be seen that using only *ae* features allows a better performance than using only *rgb*, while the highest accuracy is achieved by combining *ae* and *rgb* features. Moreover, a higher robustness w.r.t. parameters variation is achieved with *rgb-ae*. We observe that our method performs well w.r.t. sudden illumination changes even at a low learning rate, *e.g.* $\alpha = 0.03$, while the method based on only *rgb* performs more poorly at that rate (best accuracy is obtained with $\alpha = 0.10$). Using a high learning rate makes the method less robust to detecting temporary stationary objects (*e.g.* cars stopped at red traffic light). A visual comparison of the two methods performance and the ground truth mask can be observed in Fig. 4. Figure 5 shows the performance of our method w.r.t. different challenging situations. A video sequence showing the performance of our method during challenging conditions (*e.g.* rain, night) is shown online[1].

More experiments have been conducted to explore our method performance at varying AE parameters, *i.e.* number

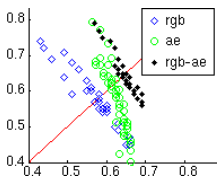

Figure 3: Overall performance (recall vs. precision)

Dec 6^{th}	D	T_b	α_T	precision	recall	F-Measure	ϵ_τ
MoG, *rgb* [32]	3	4.0	0.10	0.60	0.56	0.58	4.21 %
MoG, *ae*	128	4.0	0.10	0.62	0.58	0.60	2.89 %
MoG, *rgb-ae*	3+128	7.0	0.03	**0.66**	**0.66**	**0.66**	**2.49** %

Table 1: Best overall performance on FG detection.

Figure 5: Performance on foreground detection w.r.t. different challenges: (a) sudden light changes, (b) sudden image blurring (camera defocus), (c) night lighting with rain, (d) stationary foreground. (Left) original frame, (center) [32], (right) our method.

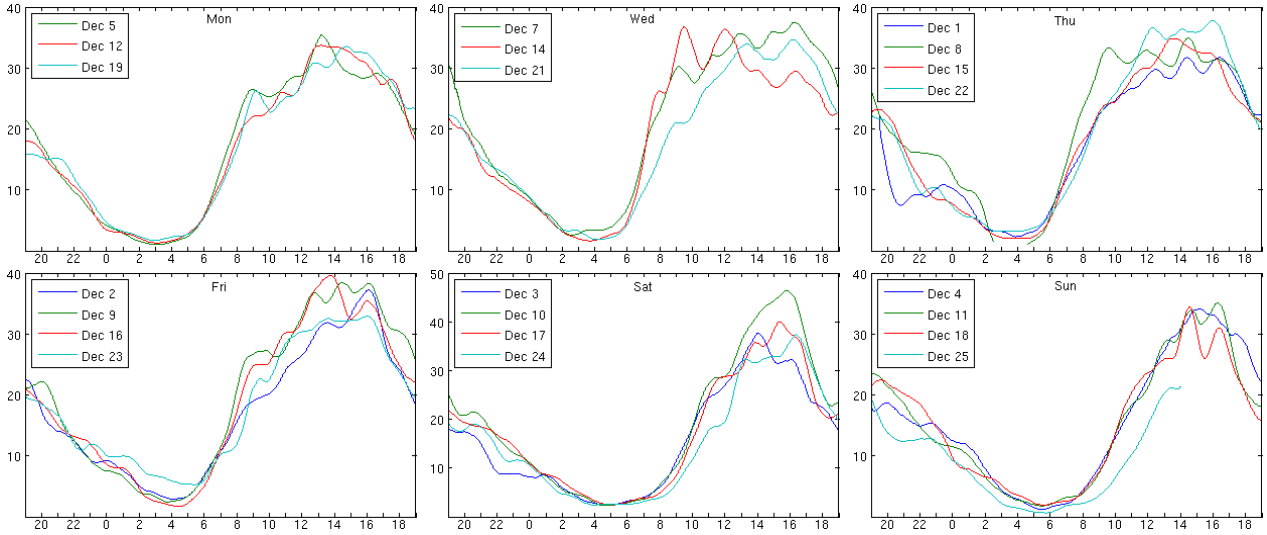

Figure 6: Patterns of behavior (average traffic intensity per time of the day) measured for the NYC-5th dataset.

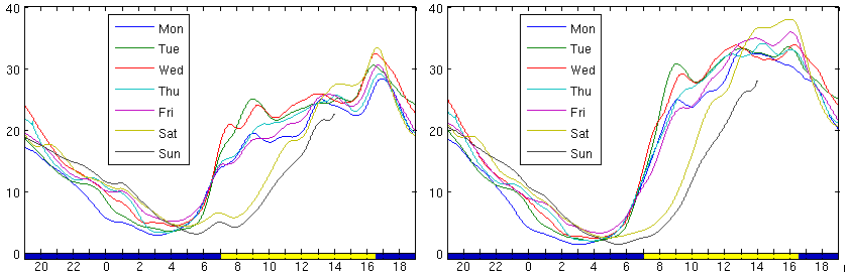

Figure 7: Typical patterns of behavior obtained for each day of the week, with (left) [32], based on *rgb* and (right) our method, based on *rgb-ae* features.

of hidden layer M and sparsity coefficient ρ. A higher performance w.r.t using only *rgb* features has been obtained with $M = 32, 64, 128$, although with $M = 128$, a higher stability has been observed w.r.t. the variation of α and T_b. The best performance is obtained with $\rho \in \{0.02, 0.04\}$, while a drop in accuracy is observed with $\rho > 0.5$. We believe that this finding is evidence of the beneficial effects of using sparse representations. Experiments have been conducted also using HOG and Gabor Filters as local features. However, the measured performance was not satisfactory. Additionally, Gabor filters requires some set up efforts, in order to select the most representative filters for the dataset (Fig.2(a)). We conjecture that the gap in performance is based on the following reason: AE method generates an over-complete dictionary, with bases having very similar but slightly shifted structures (see Fig.2(b)). This results in a sparse representation where the signal is described in terms of indices of active bins. Conversely, features like HOG or Gabor filters map patches to a dense representations, where the signal is characterized in terms of different intensities of bin values. We believe this is a key differentiation that leads to more robust representation for AE methods w.r.t. thresholding operation performed when discriminating BG from FG in our framework. Additionally, we believe that HOG features extracted at 8x8 patch level may not be able to correctly model the data variability. Our above hypothesis is in line with recent findings on sparse coding [17]. These findings suggest that while hand-designed features like SIFT or HOG perform well on the tasks for which they were initially designed, they often perform poorly on novel scenarios.

4.4 Extracting Typical and Anomalous Patterns of Behavior

The average traffic intensity measured for the whole dataset with our method is reported in Fig. 6. As the traffic intensity variability between successive frames is high, the values are smoothed in time with a median filter of 120 frames length. The behavioral pattern of Tuesday was found to be similar to the pattern of Wednesday and it was omitted for space reasons. In some cases, *e.g.* Thursday 8^{th} $4am$, the graph is interrupted because of missing data. In the next two sections, we discuss the results on typical patterns and anomalous behaviors extracted.

Typical patterns of behavior. The average typical behaviors per day of the week, obtained with [32] and with our method are shown respectively in Fig.7(a) and Fig.7(b). As highlighted in Fig.7(b), peaks in patterns due to sudden light changes, e.g. around $7am$ (sunrise) and $4{:}30pm$ (sunset), are reduced significantly. Per the findings highlighted in Fig.7(b), we observe the following: (i) Daily traffic intensity patterns are very similar to each other, *i.e.* the average traffic intensity does not vary much, given a day of the week and time of day. (ii) Two main daily behavioral patterns can be observed, one for the working days (*Mon-Fri*) and one for the weekend (*Sat*,*Sun*). For the second one, the morn-

Figure 8: Answering queries like: *How does a typical Saturday evening in New York look?* It can be easily observed that the lowest traffic density was recorded on Christmas Eve.

Figure 9: Detecting anomalous behaviors: (a) *Dec. 1^{st}, Thu, 9:00pm*, regular traffic flow is limited due to a pedestrian demonstration. (b) *Dec. 25^{th}, Sun, 9:45 am* unusual lower traffic intensity due to being Dec. 25^{th} the day of Christmas. (c) *Dec. 8^{th}, Thu, 9:30 am*, anomalous peak in traffic intensity.

ing rise in activity tends to start much later. (iii) At night, we observe an incremental drop in traffic intensity, and the average traffic intensity is sorted w.r.t. to the day of the week, going from the lowest on Sunday night, to the highest on Saturday night. Indeed, even in New York City, *the city that never sleeps*, people seem to have more bedtime before the beginning of new work weeks.

An example of a *typical Saturday night* at 9*pm* is shown in Fig.8. For each Saturday in our NYC 5th Avenue dataset, the *median frame* (i.e. the frame associated to the median τ value) within the time interval from 8:30 to 9:30*pm* is automatically extracted. We plot in Fig.8(left), the traffic intensity measured for each of the 120 frames, sorted from the lowest to the highest value. While Dec. 4^{th}, 11^{th} and 18^{th} look very similar, on Christmas Eve (Dec. 25^{th}), a lower traffic intensity is observed.

Anomalous activities. Our method can be employed for the automated detection of anomalous behavior w.r.t. the typical patterns learned. Figure 9 depicts three main anomalies detected with the method: (a) *Thu, 9pm*, unusual low traffic intensity due to the occurrence of a pedestrian demonstration on a Wednesday night and (b) *Sun, 9:45am*, unusual low traffic intensity recorded on the day of Christmas in the morning. (c) *Thu, 9:30am*, unusual high traffic intensity. Our intuition is that this traffic peak is due to a traffic congestion during to rush hours. In general, it is not always straightforward to identify the specific reason for an atypical density, but such anomalous situations can frame the search for potential explanation from the many events that occur in cities and their influences on the region at the focus of attention. In Fig.9(a) we can note that, while the behaviors on Dec. 8, 15 and 22 around 9*pm* look surprisingly similar among each others (see blue lines in the graph), the behavior on Dec.1 (see red line) differs from them remarkably. The same can be observed for Fig.9(b) and (c).A short video with the detected anomalies is shown in the supplementary material online[1].

5. CONCLUSIONS AND FUTURE WORK

We have shown how high-level patterns of behaviors can be extracted from long-term video surveillance imagery, to provide insights on the intensity of activities occurring in a city. Additionally, we showed that sparse coding applied at a *patch-* rather than *frame*-level can significantly increase the performance of foreground detection in crowded and complex urban scenarios, thus reducing the noise extracted from the imagery data. Our work is motivated by the pursuit of robust features for a stable background representation in crowded and complex scenes, and the exploration of advantages in using a sparse representation for visual surveillance.

Future directions include extending the work to a distributed visual surveillance network, in order to find correspondences among multiple cameras. Additionally, we are interested in performing a comparison of patterns of behaviors among cities or among different regions of the same city.

6. REFERENCES

[1] A. Abrams, J. Tucek, J. Little, N. Jacobs, and R. Pless. Lost: Longterm Observation of Scenes (with Tracks). *IEEE Workshop on Applications of Computer Vision (WACV)*, 2012.

[2] L. Bo, X. Ren, and D. Fox. Unsupervised feature learning for rgb-d based object recognition. *International Symposium on Experimental Robotics, (ISER)*, 2012.

[3] T. Bouwmans. Recent advanced statistical background modeling for foreground detection: A systematic survey. *Recent Patents on Computer Science*, 4(3):147–176, 2011.

[4] M. D. Breitenstein, H. Grabner, and L. V. Gool. Hunting nessie – real-time abnormality detection from webcams. *IEEE Int. Workshop on Visual Surveillance*, 2009.

[5] S. Brutzer, B. Hoferlin, and G. Heidemann. Evaluation of background subtraction techniques for video surveillance. *CVPR*, 2011.

[6] E. J. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of ACM*, 58(1):1–37, 2009.

[7] V. Cevher, C. Hegde, M. F. Duarte, and R. G. Baraniuk. Sparse signal recovery using markov random fields. *NIPS*, 2007.

[8] V. Cevher, A. Sankaranarayanan, M. Duarte, D. Reddy, R. Baraniuk, and R. Chellappa. Compressive sensing for background subtraction. *ECCV*, 2008.

[9] X. Cui, J. Huang, S. Zhang, and D. Metaxas. Background subtraction using group sparsity and low rank constraint. *ECCV*, 2012.

[10] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *CVPR*, 2005.

[11] M. Dikmen and T. S. Huang. Robust estimation of foreground in surveillance videos by sparse error estimation. *ICPR*, 2008.

[12] I. J. Goodfellow, Q. V. Le, A. M. Sav.e, H. L, and A. Y. Ng. Measuring invariance in deep networks. *NIPS*, 2009.

[13] B. Han and L. S. Davis. Density-based multifeature background subtraction with support vector machine. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34:1017–1023, 2012.

[14] N. Jacobs, N. Roman, and R. Pless. Consistent temporal variations in many outdoor scenes. *CVPR*, 2007.

[15] D. Kuettel, M. D. Breitenstein, L. V. Gool, and V. Ferrari. What's going on? Discovering spatio-temporal dependencies in dynamic scenes. *CVPR*, 2010.

[16] J. Li, S. Gong, and T. Xiang. Global behaviour inference using probabilistic latent semantic analysis. *BMVC*, 2008.

[17] J. Ngiam, C. Y. Foo, Y. Mai, C. Suen, and A. Ng. Unsupervised feature learning and deep learning tutorial. *http://ufldl.stanford.edu/wiki/index.php/UFLDL_Tutorial*.

[18] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng. Self-taught learning: Transfer learning from unlabeled data. *ICML*, 2007.

[19] V. Reddy, C. Sanderson, and B. C. Lovell. Improved foreground detection via block-based classifier cascade with probabilistic decision integration. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(1):83–93, 2013.

[20] E. Ricci, G. Zen, N. Sebe, and S. Messelodi. A prototype learning framework using EMD: Application to complex scenes analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(3):513–526, 2013.

[21] S. Rifai, Y. Bengio, A. Courville, P. Vincent, and M. Mirza. Disentangling factors of variation for facial expression recognition. *ECCV*, 2012.

[22] S. Rifai, G. Mesnil, P. Vincent, X. Muller, Y. Bengio, Y. Dauphin, and X. Glorot. Higher order contractive auto-encoder. *ECML*, 2011.

[23] R. Socher, B. Huval, B. Bhat, C. D. Manning, and A. Y. Ng. Convolutional-recursive deep learning for 3d object classification. *NIPS*, 2012.

[24] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. *CVPR*, 1999.

[25] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflowers: Principles and practise of background maintainance. *ICCV*, 1999.

[26] J. Yang, K. Yu, and T. Huang. Efficient highly over-complete sparse coding using a mixture model. *ECCV*, 2010.

[27] J. Yao and J.-M. Odobez. Multi-layer background subtraction based on color and texture. *CVPR Visual Surveillance workshop (CVPR-VS)*, 2007.

[28] G. Yu, G. Sapiro, and S. Mallat. Solving inverse problems with piecewise linear estimators: from gaussian mixture models to structured sparsity. *IEEE Transactions on Image Processing*, 21(5):2481–2499, 2012.

[29] M. Zeiler, G. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. *ICCV*, 2011.

[30] C. Zhao, X. Wang, and W. Kuen Cham. Background subtraction via robust dictionary learning. *EURASIP J. Image and Video Processing*, 2011.

[31] Y. Zhao, H. Gong, Y. Jia, and S.-C. Zhu. Background modeling by subspace learning on spatio-temporal patches. *Pattern Recognition Letters*, 2012.

[32] Z. Zivkovic. Improved adaptive gaussian mixture model for background subtraction. *ICPR*, 2004.