Adaptive Sparse Representations for Video Anomaly Detection

Xuan Mo, Student Member, IEEE, Vishal Monga, Senior Member, IEEE,
Raja Bala, Member, IEEE and Zhigang Fan, Senior Member, IEEE

Abstract—Video anomaly detection can be used in the transportation domain to identify unusual patterns such as traffic violations, accidents, unsafe driver behavior, street crime, and other suspicious activities. A common class of approaches relies upon object tracking and trajectory analysis. Very recently, sparse reconstruction techniques have been employed in video anomaly detection. The fundamental underlying assumption of these methods is that any new feature representation of a normal/anomalous event can be approximately modeled as a (sparse) linear combination pre-labeled feature representations (of previously observed events) in a training dictionary. Sparsity can be a powerful prior on model coefficients but challenges remain in: a.) the detection of anomalies involving multiple objects, and b.) the ability of the linear sparsity model to effectively allow for class separation. The proposed research addresses both these issues. First, we develop a new joint sparsity model for anomaly detection that enables the detection of joint anomalies involving multiple objects. This extension is highly non-trivial since it leads to a new simultaneous sparsity problem which we solve using a greedy pursuit technique. Second, we introduce non-linearity into, i.e. kernelize the linear sparsity model to enable superior class separability and hence anomaly detection. We extensively test on several real world video data sets involving both single and multiple object anomalies. Results show marked improvements in detection of anomalies in both supervised and unsupervised scenarios when using the proposed sparsity models.

Keywords: anomaly detection, joint sparsity model, kernel function, outlier rejection.

I. INTRODUCTION

With an increasing demand for security and safety, video-based surveillance systems are being increasingly used in urban traffic locations. Vast amounts of video footage are collected and analyzed for traffic violations, accidents, crime, terrorism, vandalism, and other suspicious activities. Since manual analysis of such large volumes of data is prohibitively costly, there is a desire to develop effective algorithms that can aid in the automatic or semi-automatic interpretation and analysis of video data for surveillance and law enforcement. An active area of research within this domain is video anomaly detection, which refers to the problem of finding patterns in data that do not conform to expected behavior, and that may warrant special attention or action.

Two precursors to anomaly detection are an effective encoding of events, and a systematic means of modeling normal events and normal event classes. The purpose of event encoding is to extract features from the video that are most useful in differentiating among different events. Xiang et al. [1] use a 7-D vector to represent each moving blob. An expectation maximization (EM) algorithm is then used to cluster these 7-D vectors into a pre-defined number of clusters. Events which can not be clustered into any of these pre-defined clusters are regarded as anomalies. A ratio histogram approach is proposed by Chuang et al. [2] to represent object features, and suspicious events such as abandoned luggage are detected using a finite state machine. Saligrama et al. [3] propose a motion label representation to encode events with decisions facilitated via a two-state Markov chain model. Wang et al. [4] propose using hierarchical Bayesian models, where the video data is divided into so-called “documents” and events are subsequently encoded as quantized features, or “words”, within these documents. Simon et al. [5] encode events via spatio-temporal volumes, and employ decision trees to identify events. In [6], the authors model shape activities of objects by a Hidden Markov Model (HMM), and define anomalies as a change in the shape activity model. Instead of extracting moving features, Malinici et al. [7] extract features of the scene as a whole and build an infinite hidden Markov model on these features to identify anomalies. An excellent review of video anomaly detection techniques can be found in [3].

Recent relevant work and challenges: Very recently, sparse reconstruction techniques [8], [9] have been employed in video anomaly detection. The fundamental underlying assumption of these methods is that any new feature representation of a normal/anomalous event can be approximately modeled as a (sparse) linear combination pre-labeled feature representations (of previously observed events) in a training dictionary. Li et al. use object trajectories while Zhao et al. use spatio-temporal volumes. Their work was motivated by the sparsity based face recognition approach of Wright et al. [10] which claimed sparse representations could exhibit robustness to significant amounts of noise and face occlusion. We note that the assertions of Wright et al. have been challenged in recent work [11], [12] and [13] which claim similar noise robustness without the use of $l_1$ norm techniques that Wright et al. advocate and [8], [9] employ. Both [8] and [9] show promise for the use of sparsity in video anomaly detection but exhibit two important limitations. First, they only address anomalies involving single objects. While such events arguably account for a large proportion of anomalies, there are important scenarios wherein the anomaly arises from an interaction among multiple objects. Consider for example, two vehicles follow-
ing nominal individual trajectories but approaching within a dangerously close vicinity of each other. The second limitation associated particularly with [8] is that the anomalous events must be characterized a priori into their own classes, i.e. supervised anomaly detection where representative training for anomalous events is available. In real world applications, it is often not possible to gather a sufficiently large number of training samples representing anomalous events.

**Contributions:** We propose a novel and general trajectory based joint sparse reconstruction framework for video anomaly detection. Trajectories have long been popular in video analysis and anomaly detection [14], [15], [16], [17]. A common characteristic of trajectory-based approaches [18]–[21] is the derivation of nominal classes of object trajectories in a training phase, and the comparison of new test trajectories against the nominal classes in an evaluation phase. A statistically significant deviation from all classes indicates an anomaly.

We must emphasize that our choice of trajectories (as opposed to spatio-temporal volumes for example in [9]) as the event encoder is motivated by two principal reasons: 1.) interactions between multiple objects are quite naturally captured in trajectory representations, e.g. vehicles approaching within a dangerously close vicinity of each other can be caught, and 2.) recent advances in object tracking ensure trajectory extraction is both fast and reliable [22]. Nevertheless, in theory any event representation can be used with our model.

*Why sparsity:* Our goal is to build a linear model where joint trajectory representations of multiple objects are written as linear combinations of corresponding joint trajectories in a training dictionary. Sparsity is a powerful prior in this model because of multiple reasons: a.) like in [8] and [9], when a new collection of multi-object trajectories manifests, it is expected to invoke only a few columns of the training dictionary that combine to create it, and b.) even more crucially object-wise correspondence is important in the linear combination for this model to physically meaningful leading to a (non-standard) block-diagonal sparse structure on coefficients detailed later in Section III-B. c.) Finally, we observe that while the sparse structure conveys information about normal/anomalous event classes - in the absence of training data for anomalous events we can develop and use outlier rejection measures on the sparse coefficient matrix that can help with multi-object anomaly detection in unsupervised settings - a very challenging problem.

Our contributions over [8] and [9] are as follows:

- **We focus on multi-object anomaly detection and extend the approaches in [8], [9] towards a joint sparsity model where a matrix (instead of a vector) of sparse coefficients results. This extension is highly non-trivial because the structure of this matrix of sparse coefficients is not naturally row sparse. The model is meaningful in the multi-object scenario only when there is object-wise correspondence in the linear combinations. To incorporate this very challenging constraint, we therefore develop and solve a new simultaneous sparsity problem with the help of a new greedy pursuit algorithm.**

- **Because linear models are not always adequate, we propose a kernelization of our joint sparsity model. If the data set does not obey linear models, kernel methods that are popular in learning, can be applied to project the data into a high-dimensional nonlinear feature space in which the data becomes more linearly separable [23]–[25]. Kernel orthogonal and basis pursuit [26], [27] algorithms and their applications [28] have been of much recent interest. We develop a new kernelization of our joint sparsity model for the purposes of video anomaly detection. This involves the development of a numerical algorithm that does not significantly increase complexity over the regular linear joint sparsity model.**

- **Finally, a suitable outlier rejection measure is developed for the multiple-object case that obviates the need to build anomalous event classes, and enables unsupervised anomaly detection with high accuracy (note, labeled training for normal events is still assumed available).**

We evaluate our algorithms by testing on several real transportation data sets for both single-object and multiple-object anomalies. Additionally, both supervised and unsupervised scenarios are used in testing. In the supervised case, training dictionaries contain both example normal and anomalous trajectories. In the unsupervised case, only normal event trajectories are available for training and the aforementioned outlier rejection measure is used for anomaly detection. The test datasets include the well known CAVIAR and AVSS [29], [30] datasets and two transportation data sets provided by Xerox Corporation. To benchmark our findings, we compare our results against widely cited trajectory based techniques. This includes: a.) the one class SVM-based method by Picarelli et al. [18], and b.) multiple-object tracking and anomaly detection approach in Han et al. [31]. For single-object anomaly detection, we also compare against the recent proposal of Li et al. [8] which was the first approach to suggest sparsity for video anomaly detection. Our experimental results include confusion matrices and ROC curves - obtained across a variety of real-world scenarios, which reveal that our proposed sparsity models outperform the alternatives.

The rest of the paper is organized as follows. Section II briefly reviews sparsity based video anomaly detection as first proposed by Li et al. Section III motivates and details our central contribution which is a joint sparsity model. This involves setting up a new simultaneous sparsity optimization problem which effectively captures representation of events as described by multiple trajectories. Section IV then presents kernelization of this joint sparsity model and proposes a new algorithm for solving the optimization problem that results from the joint kernel sparsity model (JKSM). Section V presents experimental findings. We first show the merits of sparsity under occlusion leading to missing trajectory information. Next we provide figures of merit such as confusion matrices as well as statistical evaluation via ROC Curves on real transportation video data sets to establish the virtues of our proposed approach against well-known trajectory based anomaly detection methods. Section VI concludes the paper with a discussion of possible future research directions.
which may be normal or anomalous. The trajectory representations from the training samples (i.e., example trajectory representations) from trajectory representation lie in a linear span of training trajectories (or equivalent features). Let each trajectory be approximately modeled as a (sparse) linear combination of underlying assumption in [8] is that any new trajectory can be well represented by the linear combination of trajectory no. 1 and trajectory no. 3 from class 1 (see Fig. 1). This is in fact tantamount to saying that the coefficient vector \( \alpha \) is indeed sparse - in this example, two of eight entries being active.

Based on the aforementioned model, Li et al. [8] identify anomalies by apriori defining normal and anomalous event classes in video. Their advocacy of sparsity over other trajectory classification techniques viz. one-class SVMs [18] is based on two arguments: 1.) recent work in face recognition [10] has shown that sparsity based classification can be powerful even as feature descriptions are missing, e.g., occlusion of objects leading to missing trajectory information, and 2.) the sparse coefficients can additionally withstand noise and other image distortions. The claim of noise/distortion robustness, particularly of \( \ell_1 \) type sparse representations has been challenged recently [11], [12] and [13].

Nevertheless, in models based on a training dictionary, sparsity is a powerful prior particularly in multi-object scenarios where the structure of sparse coefficients in a matrix can be very instrumental in both normal vs. anomalous event classification as well as developing outlier rejection measures. The next Section details our first contribution.

## II. Review of Sparsity-Based Anomaly Detection

Abnormal behavior detection via sparse reconstruction analysis of trajectory [8] is a recent novel and promising idea in the field of video anomaly detection. The fundamental underlying assumption in [8] is that any new trajectory can approximately be modeled as a (sparse) linear combination of training trajectories (or equivalent features). Let each trajectory representation lie in \( \mathbb{R}^T \), and \( T \) denote the number of training samples (i.e., example trajectory representations) from each of \( K \) different classes, i.e., behavior patterns in a video which may be normal or anomalous. The \( T \) training samples (trajectory representations) from the \( i \)-th class are arranged as the columns of a matrix \( A_i \in \mathbb{R}^{n \times T} \). The dictionary \( A \in \mathbb{R}^{n \times KT} \) of training samples from all classes is then formed as follows:

\[
A = [A_1 \ A_2 \ldots A_K].
\]

Given a sufficient number of training samples from the \( m \)-th trajectory class, a test image \( y \in \mathbb{R}^n \) from the same class is conjectured to approximately lie in the linear span of those training samples. Any trajectory feature vector is synthesized by a linear combination of the set of all training trajectory samples as follows:

\[
y \approx A\alpha = [A_1 \ A_2 \ldots A_K] \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_K \end{bmatrix},
\]

(1)

where each \( \alpha_i \in \mathbb{R}^T \). Typically for an example trajectory \( y \), only one of the \( \alpha_i \)'s will be active (corresponding to the class/event from which \( y \) is generated). Thus the coefficient vector \( \alpha \in \mathbb{R}^{KT} \) is modeled as sparse and is recovered by solving the following optimization problem:

\[
\hat{\alpha} = \arg \min_{\alpha} \| \alpha \|_1 \text{ subject to } \| y - A\alpha \|_2 < \epsilon,
\]

(2)

where the objective is to minimize the number of non-zero elements in \( \alpha \). The \( \ell_1 \) norm can capture sparsity in a convex set up [32] but has shown to be quite expensive [11]. The residual error between the test trajectory and each class behavior pattern is computed to find the class to which the test trajectory belongs:

\[
r_i(y) = \| y - A_i\hat{\alpha} \|_2 \quad i = 1, 2, \ldots, K
\]

(3)

Fig. 1 shows an example of classification using sparsity model. The training dictionary consist of 2 classes, each class contains 4 different trajectories. The test trajectory can be very instrumental in both normal vs. anomalous event classes in video. Their advocacy of sparsity over other trajectory classification techniques viz. one-class SVMs [18] is based on two arguments: 1.) recent work in face recognition [10] has shown that sparsity based classification can be powerful even as feature descriptions are missing, e.g., occlusion of objects leading to missing trajectory information, and 2.) the sparse coefficients can additionally withstand noise and other image distortions. The claim of noise/distortion robustness, particularly of \( \ell_1 \) type sparse representations has been challenged recently [11], [12] and [13].

Nevertheless, in models based on a training dictionary, sparsity is a powerful prior particularly in multi-object scenarios where the structure of sparse coefficients in a matrix can be very instrumental in both normal vs. anomalous event classification as well as developing outlier rejection measures. The next Section details our first contribution.

## III. A Joint Sparsity Model for Trajectory Based Video Anomaly Detection

### A. Observations and Motivation for a Joint Sparsity Model

A variety of anomaly detection algorithms have been designed for video surveillance. However only few of them have considered the interaction between multiple objects [4], [6], [31]. While it is true that anomalies are generated by atypical trajectory/behavior of a single object, “collective anomalies” that are caused by the joint observation of objects are also significant. For example, in the area of transportation, some events, e.g., accidents and dangerous driver-pedestrian behavior, are indeed based on joint and not just individual object behavior. Previous methods have employed probabilistic models to learn the relationship between different individual events. Han et al. [31] and Vaswani et al. [6] use an HMM-based method to track multiple trajectories followed by defining a set of rules to distinguish between normal and anomalous events. Wang et al. [4] present an unsupervised framework using hierarchical Bayesian models to model individual events and interactions between them.

The sparsity based approach reviewed in Section II while powerful, does not capture interactions to detect 2 or more object anomalies. We describe next a new “joint sparsity model” for video anomaly detection which incorporates multiple object trajectories and their interactions. Hence even if individually the trajectories may be considered normal, “collective anomalies” could occur and can be successfully detected in our proposed framework.
B. A Joint Sparsity Model for Anomaly Detection

We are interested in detection of anomalies involving \( P \geq 1 \) objects. Their corresponding \( P \) trajectories can be represented as a matrix: \( \mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_P] \in \mathbb{R}^{n \times P} \), where \( \mathbf{y}_i \) correspond to \( i^{th} \) trajectory. The training dictionary can be defined as: \( \mathbf{A} = [\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_P] \in \mathbb{R}^{n \times PKT} \), where each dictionary \( \mathbf{A}_i = [\mathbf{A}_{i,1}, \mathbf{A}_{i,2}, \ldots, \mathbf{A}_{i,K}] \in \mathbb{R}^{n \times KT}, i = 1, 2, \ldots, P \), is formed by the concatenation of the sub-dictionaries from all classes belonging to the \( i^{th} \) trajectory. The crucial aspect of this formulation is that the training trajectories for any class \( j \), i.e. \( \mathbf{A}_{i,j}, i = 1, 2, \ldots, P \) are observed “jointly” from example videos. This generalizes the set-up of [8], [9].

The test \( P \) trajectories can now be represented as a linear combination of training samples as follows:

\[
\mathbf{Y} = \mathbf{A} \mathbf{S} = [\mathbf{A}_{1,1} \mathbf{A}_{1,2} \ldots \mathbf{A}_{1,K} \ldots \mathbf{A}_{P,1} \mathbf{A}_{P,2} \ldots \mathbf{A}_{P,K}] [\mathbf{a}_1 \ldots \mathbf{a}_P] \tag{4}
\]

where the coefficient vectors \( \mathbf{a}_i \) lie in \( \mathbb{R}^{PKT} \) and \( \mathbf{S} = [\mathbf{a}_1 \ldots \mathbf{a}_i \ldots \mathbf{a}_P] \).

It is important to note that the \( i^{th} \) object trajectory of any observed set of test trajectories should only lie in the span of training trajectories corresponding to the \( i^{th} \) object. Therefore, the columns of \( \mathbf{S} \) should have the following structure:

\[
\mathbf{a}_1 = \begin{bmatrix} \alpha_{1,1} \\ \alpha_{1,2} \\ \vdots \\ \alpha_{1,K} \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{a}_i = \begin{bmatrix} 0 \\ \alpha_{i,1} \\ \alpha_{i,2} \\ \vdots \\ \alpha_{i,K} \\ 0 \end{bmatrix}, \quad \mathbf{a}_P = \begin{bmatrix} 0 \\ \alpha_{P,1} \\ \alpha_{P,2} \\ \vdots \\ \alpha_{P,K} \\ 0 \end{bmatrix} \tag{5}
\]

where each of the sub-vectors \( \{\alpha_{i,j}\}_{j=1}^K, i = 1, 2, \ldots, P \) lies in \( \mathbb{R}^T \), while \( \mathbf{0} \) denotes a vector of all zeros in \( \mathbb{R}^{KT} \). As a result, \( \mathbf{S} \) exhibits a block-diagonal structure.

From [8], we know that for a single object, its trajectory can be represented by a sparse linear combination of all the training samples. For the multiple trajectories scenario, we assume that training samples with non-zero weights (in the sparse linear combination) exhibit one-one correspondence across different trajectories. In other words, if the \( i^{th} \) trajectory training sample from the \( j^{th} \) class is chosen for the \( i^{th} \) test trajectory, then it is necessarily that other \( P - 1 \) trajectories choose from \( j^{th} \) class with very high probability, albeit with possibly different weights.

We take a simple scenario which only has 2 objects and 2 training classes (normal and anomalous class) as an example to explain the structure of Eq. (4). In this situation, \( P = 2, K = 2 \), Eq. (4) becomes:

\[
\mathbf{Y} = \mathbf{A} \mathbf{S} = [\mathbf{A}_{1,1} \mathbf{A}_{1,2} \mathbf{A}_{2,1} \mathbf{A}_{2,2}] \begin{bmatrix} \alpha_{1,1} & 0 & \alpha_{1,2} & 0 \\ 0 & \alpha_{2,1} & 0 & \alpha_{2,2} \end{bmatrix} \tag{6}
\]

The test trajectory sample is thought of as a collective event. Therefore, all trajectories of the sample should be classified into one class. If the \( 1^{st} \) trajectory is classified into \( j^{th} \) class, the \( 2^{nd} \) trajectory should also be classified into \( j^{th} \) class, which means \( \mathbf{a}_{1,j} \) and \( \mathbf{a}_{2,j} \) should be activated simultaneously. This characteristic that some coefficients should be activated jointly captures the interaction between objects.

Moreover, we only care about the non-zero element in the matrix \( \mathbf{S} \). Define a new matrix \( \mathbf{S}' \):

\[
\mathbf{S}' = \begin{bmatrix} \alpha_{1,1} & \alpha_{1,2} & \ldots & \alpha_{1,P} \\ \alpha_{1,2} & \alpha_{1,2} & \ldots & \alpha_{1,P} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{K,1} & \alpha_{K,2} & \ldots & \alpha_{K,P} \end{bmatrix} \tag{7}
\]

In the structure of \( \mathbf{S}' \), “joint coefficients” are moved into the same row. The joint information can be captured by enforcing certain entire rows of \( \mathbf{S}' \) to be activated simultaneously.

In general, when there are \( K \) classes and \( P \) objects, the structure of \( \mathbf{S}' \) is:

\[
\mathbf{S}' = \begin{bmatrix} \alpha_{1,1} & \ldots & \alpha_{1,1} & \ldots & \alpha_{1,P} \\ \alpha_{1,2} & \ldots & \alpha_{1,2} & \ldots & \alpha_{1,P} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_{K,1} & \ldots & \alpha_{K,1} & \ldots & \alpha_{K,P} \end{bmatrix} \in \mathbb{R}^{KT \times P}. \tag{8}
\]

The question that remains to be addressed is the particular way of transforming \( \mathbf{S} \). Such a transformation is realized by defining matrices \( \mathbf{H} \in \mathbb{R}^{PKT \times P} \) and \( \mathbf{J} \in \mathbb{R}^{KT \times PKT} \):

\[
\mathbf{H} = \begin{bmatrix} 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 1 \end{bmatrix}, \quad \mathbf{J} = \begin{bmatrix} \mathbf{I}_{KT} & \mathbf{I}_{KT} & \ldots & \mathbf{I}_{KT} \end{bmatrix}. \tag{9}
\]

The vectors \( \mathbf{1} \) and \( \mathbf{0} \) are in \( \mathbb{R}^{KT} \) and contain all ones and zeros respectively, and \( \mathbf{I}_{KT} \) is the \( KT \)-dimensional identity matrix.

Then, we have:

\[
\mathbf{J}(\mathbf{H} \circ \mathbf{S}) = \mathbf{S}', \tag{11}
\]

where the \( \circ \) indicates matrix Hadamard (entry-wise) product.

Therefore, we can now solve for the sparse coefficients via the following optimization problem:

\[
\text{minimize } \| \mathbf{J}(\mathbf{H} \circ \mathbf{S}) \|_{\text{row,0}} \quad \text{subject to } \| \mathbf{Y} - \mathbf{A} \mathbf{S} \|_F \leq \varepsilon, \tag{12}
\]

where \( \| \cdot \|_{\text{row,0}} \) refers to the number of non-zero rows in the matrix and the cost function minimization seeks the \( \mathbf{J}(\mathbf{H} \circ \mathbf{S}) \) with the minimum number of non-zero rows, while the constraint ensures good approximation (\( \| \cdot \|_F \) denotes the Frobenius norm). It is worth re-emphasizing that the matrix \( \mathbf{S}' \) has a special structure - elements in the same row should be activated simultaneously which is captured by using the matrix \( \| \cdot \|_{\text{row,0}} \) norm. Enforcing sparsity of \( \mathbf{S}' \) (equivalently \( \mathbf{S} \)) using other traditional matrix norms, e.g. entry-wise \( l_p \) norms, can be detrimental to performance [33], [34] particularly under the quadratic reconstruction constraint because they could lead to solutions that depart significantly from row sparsity.
The well-known row sparsity problem:

\[
\text{minimize } ||S||_{\text{row},0} \\
\text{subject to } ||Y - AS||_F \leq \varepsilon,
\]

(13)
is non-convex but can be solved using greedy pursuit algorithms widely used in the literature. Simultaneous Orthogonal Matching Pursuit (SOMP) [34], [35] - enumerated in Algorithm 1 - is amongst the most popular algorithms used. In SOMP, the support of the solution is sequentially updated (i.e., the atoms in the dictionary \( A \) are sequentially selected). At each iteration, the atom that simultaneously yields the best approximation to all of the residual vectors is selected.

\[
\lambda_k = \arg \max_i ||R_{k-1}^j a_i||_2
\]

(14)

Our proposed joint sparsity model for representing multiple object trajectories involves solving Eq. (12), which looks the general formulation even when the Hadamard operator is effectively gives

\[
\text{after employing this special rule of choice for atom selection,}
\]

each row of parameter matrix \( A \) and creates an orthogonal projection with the highest correlation.

Algorithm 1 SOMP

Input: Dictionary \( A = [a_1, a_2, \ldots, a_{PKT}] \), data matrix \( Y = [y_1, y_2, \ldots, y_P] \), the stopping criterion: \( \frac{||Y - A\Lambda_{k-1}S_k||_F}{||Y - A\Lambda_{k-1}S_k||_F} > 1 - \mu \), where \( \mu \) is a small positive number

1: initialization: residual \( R_0 = Y \), index set \( \Lambda_0 \): empty set, iteration counter \( k = 1 \)
2: while stopping criterion has not been met do
   1) Find the index of the atom that best approximates all residuals: \( \lambda_k = \arg \max_i ||R_{k-1}^j a_i||_2 \)
   2) Update the index set \( \Lambda_k = \Lambda_{k-1} \cup \{\lambda_k\} \)
   3) Compute \( G_k = (A_{\Lambda_k}^TA_{\Lambda_k})^{-1}A_{\Lambda_k}^TY \). \( A_{\Lambda_k} \) consists of the \( k \) atoms in \( \Lambda \) indexed in \( \Lambda_k \)
   4) Determine the residual \( R_k = Y - A_{\Lambda_k}^T G_k \)
   5) \( k \leftarrow k + 1 \)
3: end while

Output: Index set \( \Lambda = \Lambda_{k-1} \), the sparse representation \( S \) whose nonzero rows indexed by \( \Lambda \) are \( k \) rows of the matrix \( (A_{\Lambda_k}^TA_{\Lambda_k})^{-1}A_{\Lambda_k}^TY \)

This idea can be extended to our proposed joint sparsity setting. If the atom of \( j \)-th trajectory we selected comes from \( i \)-th training, the other \( P - 1 \) atoms of trajectories should be also chosen from \( i \)-th training. Then Eq. (14) in SOMP can be modified as:

\[
\lambda_k = \arg \max_i \sum_j ||R_{k-1}^{j,j} a_{j,i}||_2
\]

(15)

where \( R_{k-1}^{j,j} \) refers to the residual of \( j \)-th trajectory in iteration \( k - 1 \), and \( a_{j,i} \) represents the \( i \)-th training of \( j \)-th trajectory. After employing this special rule of choice for atom selection, each row of parameter matrix \( S' \) will be activated simultaneously or inactivated simultaneously, thus the row sparsity requirements will inherently hold. The implementation details of this algorithm can be found in a technical report [36].

Supervised Anomaly Detection as Event Classification

Having obtained the sparse coefficient matrix \( S \), we compute class-specific residual errors and identify the class of the test event \( Y \) as that which gives the minimum residual:

\[
\text{identity}(Y) = \arg \min_i ||Y - AS_i||_F,
\]

(16)

where \( \delta_i(S) \) is the matrix whose only nonzero entries are the same as those in \( S \) associated with class \( i \) (in all \( P \) trajectories). When sufficient representation (example training trajectories) for anomalous events is available, then anomalous classes are simply one or more of the \( K \) classes in this joint sparsity based classification framework.

Unsupervised Anomaly Detection via Outlier Rejection

If training for anomalies is missing/statistically insignificant, we can not use (16) to identify anomalies. Inspired by the outlier rejection measure in [10]:

\[
SCI(\alpha) = \frac{K \cdot \max_i ||p_i(\alpha)||_1/||\alpha||_1 - 1}{K - 1}
\]

(17)

and \( p_i(\alpha) \) is the new vector whose only nonzero entries are the entries in \( \alpha \) that are associated with class \( i \). We model anomalies as outliers, given training from expected normal event classes that form the dictionary \( A \). Eq. (17) can be used to detect single object anomalies. We extend it to the multiple object case:

\[
JSCI(S') = \frac{K \cdot \max_i ||\delta_i(S')||_{\text{row},0}/||S'||_{\text{row},0} - 1}{K - 1},
\]

(18)

where \( JSCI \) is the Joint-SCI. Note that \( 0 \leq JSCI(S') \leq 1 \), if \( JSCI(S') \) is close to 0, the event is normal, and if \( JSCI(S') \) is close to 1, the event is anomalous. We choose the threshold \( \tau_2 \in (0,1) \), if \( JSCI(S') < \tau_2 \), a multiple object anomaly is identified. A nominal choice of \( \tau_2 = 0.5 \) can be made but this can further be optimized experimentally based on the underlying video data set by observing the range of the measure \( JSCI(S') \) for normal events.

Fig. 2 shows an example of sparse coefficients under normal...
vs. anomalous events. To the left of the figure are sparse coefficients of a anomalous event. The activated coefficients are scattered all over the normal classes, therefore the corresponding event can not be classified into any class. The JSCI value in Eq. (18) will therefore be small. On the other hand, the activated coefficients in the right figure are clustered in normal class 2. Correspondingly, the JSCI value will be large.

IV. KERNEL SPARSITY FOR TRAJECTORY BASED VIDEO ANOMALY DETECTION

The effectiveness of the proposed joint sparsity model largely depends on the structure of the trajectory data. If the data is not linearly separable enough, the trajectory-based sparsity model may not enable sufficiently accurate reconstruction to be reliable from a classification standpoint. Kernel methods can be applied to transform the data into a feature space via a transformation \( \phi(x) \) such that the resulting transformed trajectory vectors becomes more separable \([23, 25]\) and comply with the linear sparsity model.

The kernel function \( \kappa: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R} \), is usually defined as the inner product:
\[
\kappa(x, z) = \langle \phi(x), \phi(z) \rangle
\]

The transformed training trajectory vectors are written as:
\[
a_i \mapsto \phi(a_i), \quad \text{where } a_i \text{ is the } i\text{-th column of } A.
\]

The transformed trajectory vectors are written as:
\[
\phi(y) = \begin{bmatrix} \phi(a_1) & \cdots & \phi(a_{K^T}) \end{bmatrix} \begin{bmatrix} \alpha'_1 \cdots \alpha'_{K^T} \end{bmatrix}^T = A_{\phi} \alpha',
\]

where \( \alpha' \) is also assumed to be sparse.

Similar to Eq. (2), the new sparse coefficient vector \( \alpha' \in \mathbb{R}^{K^T} \) can be recovered by solving:
\[
\hat{\alpha}' = \arg\min_{\alpha'} \| \alpha' \|_1 \text{ subject to } \| \phi(y) - A_{\phi} \hat{\alpha}' \|_2 < \varepsilon,
\]

The problem in Eq. (21) can be approximately solved by kernel orthogonal/basis matching pursuit algorithms \([26, 28]\). Note that in the above problem formulation, we are solving for the sparse vector \( \alpha' \) directly in the feature space using the implicit feature vectors, but not evaluating the kernel functions at the training points.

The effectiveness of the proposed joint sparsity model largely depends on the structure of the trajectory data. If the data is not linearly separable enough, the trajectory-based sparsity model may not enable sufficiently accurate reconstruction to be reliable from a classification standpoint. Kernel methods can be applied to transform the data into a feature space via a transformation \( \phi(x) \) such that the resulting transformed trajectory vectors becomes more separable \([23, 25]\) and comply with the linear sparsity model.

The kernel function \( \kappa: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R} \), is usually defined as the inner product:
\[
\kappa(x, z) = \langle \phi(x), \phi(z) \rangle
\]

The transformed training trajectory vectors are written as:
\[
a_i \mapsto \phi(a_i), \quad \text{where } a_i \text{ is the } i\text{-th column of } A.
\]

The transformed trajectory vectors are written as:
\[
\phi(y) = \begin{bmatrix} \phi(a_1) & \cdots & \phi(a_{K^T}) \end{bmatrix} \begin{bmatrix} \alpha'_1 \cdots \alpha'_{K^T} \end{bmatrix}^T = A_{\phi} \alpha',
\]

where \( \alpha' \) is also assumed to be sparse.

Similar to Eq. (2), the new sparse coefficient vector \( \alpha' \in \mathbb{R}^{K^T} \) can be recovered by solving:
\[
\hat{\alpha}' = \arg\min_{\alpha'} \| \alpha' \|_1 \text{ subject to } \| \phi(y) - A_{\phi} \hat{\alpha}' \|_2 < \varepsilon,
\]

The problem in Eq. (21) can be approximately solved by kernel orthogonal/basis matching pursuit algorithms \([26, 28]\). Note that in the above problem formulation, we are solving for the sparse vector \( \alpha' \) directly in the feature space using the implicit feature vectors, but not evaluating the kernel functions at the training points.

Algorithm 2 KSOMP

**Input:** Dictionary \( A = [a_1, a_2, \ldots, a_{PKT}] \), data matrix \( Y = [y_1, y_2, \ldots, y_P] \), kernel function \( \kappa \), the stopping criterion:
\[
\| y - A_{\phi} S_0 \|_F \geq 1 - \mu
\]

1: **initialization:** compute the kernel matrices \( \Theta_A \in \mathbb{R}^{PKT \times PKT} \), whose \((i,j)\)-th entry is \( \kappa(a_i, a_j) \) and \( \Theta_{A,Y} \in \mathbb{R}^{PKT \times P} \), whose \((i,j)\)-th entry is \( \kappa(a_i, y_j) \). Set index set \( \Lambda_0 = \arg\max_i \| \Theta_{A,Y} \|_i \).
2: **while** stopping criterion has not been met **do**
   1: Compute the correlation matrix
   \[
   C = \Theta_{A,Y} - (\Theta_A)_{A=1} \Theta_{A,Y} (\Theta_A)_{A=1} + \lambda I
   \]
   2: Select the new index as \( \lambda_t = \arg\max_i \| C_i \|_2 \)
   3: Update the index set \( \Lambda_t = \Lambda_{t-1} \cup \{ \lambda_t \} \)
   4: \( t \leftarrow t + 1 \)
3: **end while**

**Output:** Index set \( \Lambda = \Lambda_{t-1} \), the sparse representation \( S_0 \) whose nonzero rows indexed by \( S_0 = (\Theta_{A,A} + \lambda I)^{-1}(\Theta_{A,Y})_{\Lambda} \).

The well-known row sparsity problem in Eq. (13) can be extended to the (kernelized) feature space as follows:
\[
\begin{align*}
\text{minimize} & \quad \| S_0 \|_{row,0} \\
\text{subject to} & \quad \| Y_0 - A_{\phi} S_0 \|_F \leq \varepsilon,
\end{align*}
\]

We propose the kernel SOMP algorithm, i.e. KSOMP in order to solve Eq. (21) - see Algorithm 2. Note that a regularization term \( \lambda I \) is added in Step 1 of Algorithm 2 order to enable a stable inversion. Like SOMP (Algorithm 1), the goal of KSOMP is to pick out non-zero rows that minimize \( \| S_0 \|_{row,0} \) but in the transformed kernel space.

The kernelized version of our proposed joint sparsity model in Eq. (12) is given as:
\[
\begin{align*}
\text{minimize} & \quad \| J (H \circ S_0) \|_{row,0} \\
\text{subject to} & \quad \| Y_0 - A_{\phi} S_0 \|_F \leq \varepsilon,
\end{align*}
\]

which is very similar to Eq. (21) but for the presence of the matrices \( J, H \) and the Hadamard operator. Recall the modification to SOMP employed in Section III a similar trick will allow us to adapt kernelized SOMP to yield a solution for our problem in Eq (22). In particular, instead of using the selection rule:
\[
\lambda_t = \arg\max_i \| C_i \|_2,
\]

in Step 2 of KSOMP, we can jointly select atoms of trajectories from the the same training:
\[
\lambda_t = \arg\max_i \sum_j \| C_{ij} \|_2,
\]

where \( C_{ij} \) refers to the correlation matrix of \( j\)-th trajectory.
Note that in the transformed space, the residual becomes:
\[
\| \phi(y) - A_0 \alpha \|^2_2 = \left[ \sum_{i=1}^{n} (\phi(y)_i - (A_0 \alpha'_{i})))^2 \right]^{1/2} \\
= \left[ \sum_{i=1}^{n} (\phi(y)_i - \sum_{j=1}^{KT} \alpha'_{j}(A_0 \alpha'_{i},j))^2 \right]^{1/2} \\
= \left[ \sum_{i=1}^{n} \phi(y)_i - 2 \phi(y)_i \sum_{j=1}^{KT} \alpha'_{j} + \sum_{j=1}^{KT} \alpha'_{j} \right]^{1/2} \\
= \left[ \sum_{i=1}^{n} \phi(y)_i - 2 \sum_{j=1}^{KT} \phi(y)_i (A_0 \alpha'_{i},j) \right]^{1/2} \\
= \left( \kappa(y,y) - 2 \alpha^T \theta_{A \gamma} - \alpha^T \Theta_{A \alpha} \right)^{1/2}
\]

Let \( \hat{S}_0 \) be the optimum solution of Eq. \( \ref{eq:22} \). The residual for the kernelized joint sparsity model corresponding to the \( i \)-th class is then given by:
\[
\| Y_0 - A_0 \delta_i(\hat{S}_0) \|_F = \sum_{j=1}^{p} \left( \| \phi(y)_j - A_0 (\delta_i(\hat{S}_0)) \|_2 \right)^2, \quad (25)
\]
where \( \phi(y)_j \) is the \( j \)-th column of \( Y_0 \) and \( (\delta_i(\hat{S}_0))_j \) is the \( j \)-th column of \( \delta_i(\hat{S}_0) \). More, precisely in terms of the kernel function, this is given by:
\[
r_i(Y_0) = \left( \sum_{j=1}^{p} \kappa(y_j,y_j) - 2 \delta_i(\hat{S}_0) \right)^T (\Theta_{A \gamma} \delta_i(\hat{S}_0)) + \left( \delta_i(\hat{S}_0) \right)^T (\Theta_{A \alpha} \delta_i(\hat{S}_0))^{1/2},
\]
where \( \Omega_i \) is the index set associated with the \( i \)-th training class.

**Supervised Anomaly Detection as Event Classification**

Similar to Eq. \( \ref{eq:16} \), the class of \( Y_0 \) is determined by:
\[
\text{identity}(Y_0) = \arg \min_{i} \| Y_0 - A_0 \delta_i(\hat{S}_0) \|_F, \quad (26)
\]
where \( \delta_i(\hat{S}_0) \) is the matrix whose only nonzero entries are the same as those in \( \hat{S}_0 \) associated with class \( i \).

**Unsupervised Anomaly Detection via Outlier Rejection**

As in Section \( \ref{sec:3} \) we can define a transformed coefficient matrix \( \hat{S}_0 = J(H \circ \hat{S}_0) \). The outlier rejection measure in Eq. \( \ref{eq:18} \) can then be extended here as:
\[
JSCI(\hat{S}_0) = \frac{K \cdot \max_{i} \| \delta_i(\hat{S}_0) \|_{row,0} / \| \hat{S}_0 \|_{row,0} - 1}{K - 1}, \quad (27)
\]

**Kernel Parameters Optimization**

We focus on picking parameters for the RBF kernel \( \kappa(x,z) = e^{-\gamma\|x-z\|^2} \) which will be used in all our experiments.

For different choices of the RBF parameter \( \gamma \), multiple training dictionaries are generated \( A_0(\gamma) \), i.e. \( A_0(\gamma) \) is a function of \( \gamma \). Inspired by cross-validation, we split the training data \( A_0(\gamma) \) into two subsets \( B_0(\gamma) \) and \( C_0(\gamma) \), such that both \( B_0(\gamma) \) and \( C_0(\gamma) \) have representation from the \( K \) classes. Now if the dictionary in the sparsity model is chosen to be equal to \( B_0(\gamma) \) and a (transformed) test trajectory is picked from \( C_0(\gamma) \), then *ideally* we expect perfect classification into one of the \( K \) classes. Therefore, a good kernel is one that will enable close to ideal classification of test samples from \( C_0(\gamma) \) - which means that only a small number of \( \hat{S}_0(\gamma) \) are activated (non-zero) and for one particular class. Here the outlier rejection measure:
\[
JSCI(\hat{S}_0(\gamma)) = \frac{K \cdot \max_{i} \| \delta_i(\hat{S}_0(\gamma)) \|_{1} / \| \hat{S}_0(\gamma) \|_{1} - 1}{K - 1}, \quad (28)
\]
where \( \delta_i(\hat{S}_0(\gamma)) \) is the vector whose only nonzero entries are the same as those in \( \hat{S}_0(\gamma) \) associated with class \( i \). \( JSCI(\hat{S}_0(\gamma)) \) will be very close to 1 if the classification is accurate. Therefore the best parameter \( \gamma \) can be chosen by solving the following kernel parameter optimization problem:
\[
\arg \max_{\gamma} JSCI(\hat{S}_0(\gamma)), \quad (29)
\]

**Computational Complexity**

Suppose that the dimension of trajectory is \( n: \gamma \in \mathbb{R}^n \). For the SOMP Algorithm \( \ref{alg:SOMP} \) the complexity is \( O(nKPT) \) in our set up - where \( K, P, T \) were defined in Section \( \ref{sec:3} \) and refer to the number of classes, objects and training samples per class respectively. Changing the selection rules in our modified version of SOMP will not increase the complexity much. So, the computational complexity of our joint sparsity model is also \( \approx O(nKPT) \). The kernelized joint sparsity model has to compute the kernel matrix \( \Theta_{A} \) and the correlation matrix \( C \). The computational complexity will therefore increase to \( O(nKP^2T) \). Note that the number of objects \( P \) is usually a small number. Therefore, kernel trick does not significantly increase complexity over sparsity based anomaly detection while providing significant performance improvements as will be seen shortly in Section \( \ref{sec:5} \).

**V. Experimental Validation**

**A. Trajectory Extraction**

Trajectory extraction is accomplished using well-known techniques. First background subtraction is accomplished via the use of a Gaussian Mixture Model (GMM) \( \ref{fig:4} \). In order to eliminate the effect of noise, blob analysis is then used here to identify the location of the moving vehicle. We calculate the number of connected foreground pixels, and deem the connected segment to be a vehicle if this exceeds a threshold. As seen in Fig. \( \ref{fig:4} \), a), the car is successfully detected by this technique. Next, we calculate and track the centroid of the blob over time in order to obtain the object trajectory. Fig. \( \ref{fig:4} \)b) shows an example of the extracted trajectory, which is represented mathematically as a coordinate pair \( \{x(t), y(t)\} \).

Li et al. \( \ref{fig:8} \) use a LCSCA (Least-squares Cubic Spline Curves Approximation) representation of trajectories. We first approximate a raw trajectory using a B-spline function \( \ref{fig:8} \) with 50 knots (50 x-coordinates and 50 y-coordinates) and these knots are extracted to represent the trajectory.

**B. Brief Review of Competing Approaches**

Here we elaborate on three widely cited techniques from the literature that will be used as benchmarks to compare our approach.
calculation and trajectory derivation by collecting the blob centroid.

Fig. 4. Trajectory extraction: (a) background subtraction using gaussian mixture models and blob analysis to identify objects, (b) blob centroid calculation and trajectory derivation by collecting the blob centroid.

Fig. 5. Illustration of one class SVM method by Piciarelli et al. 

Anomaly Detection via SVM Trajectory Clustering

Piciarelli et al. [18] propose a technique that uses one class SVMs for anomaly detection by utilizing trajectory information as features.

In their method, trajectories are represented using 8 pairs of \( x \) and \( y \) coordinates, thus leading to feature vectors composed of 16 elements. Then, these trajectories are clustered with a one-class SVM. During the training phase, a classification hyperplane is learned. Fig. 5 shows an illustration of their method on 2-D data. The classification hyperplane intersects the hypersphere thus defining an hyperspherical cap containing the majority of the data, while outliers lie outside the cap.

Online Anomaly Detection using Sparsity Model

Zhao et al. [9] propose an online sparsity-based method for video anomaly detection. They use sparse linear combinations of spatio-temporal volumes under an \( l_1 \) sparsity model. But instead of using a fixed training dictionary and only solving for the sparse coefficients, they employ a principled convex optimization formulation that allows both a sparse reconstruction code, and an online dictionary to be jointly inferred and updated:

\[
(\alpha_1, \ldots, \alpha_m, A^\ast) = \arg \min_{\alpha_1, \ldots, \alpha_m, A} \sum_{i=1}^{m} J(y_i, \alpha_i, A),
\]

where \( J(y_i, \alpha_i, A) \) is the objective function that measures how normal an event is. It includes reconstruction error, sparsity regularization via \( l_1 \) norm and smoothness terms. Subsequent to the optimization, the dictionary \( A \) is augmented using newly observed events.

For anomaly detection, they define a threshold \( \hat{e} \) which that controls the sensitivity of the algorithm to anomalous events. A test spatio-temporal volume \( y' \) will be detected as anomalous event if the following criterion is satisfied:

\[
J(y', \alpha', A^\ast) > \hat{e}
\]

Our implementation of Zhao’s method is based on their pseudocode in Algorithm 1 on Page 4 of their paper [9].

Multi-Object Trajectory Tracking and Anomaly Detection

Han et al. [31] propose a multiple object tracking algorithm and corresponding rule-based anomaly detection approach. In their tracking algorithm, each object is identified by index \( i \) and its state at time \( t \) is represented by:

\[
x_i^t = (p_i^t, v_i^t, a_i^t, s_i^t)
\]

where \( p_i^t \) is the image location, \( c_i^t \) represents the 2D velocity, \( a_i^t \) and \( s_i^t \) denote the appearance and scale of object \( i \) at time \( t \), respectively. \( p_i^t \) and \( v_i^t \) use continuous image coordinates. Then, they use a Hidden Markov Model as the probabilistic model to maximize the joint probability between the state sequence and the observation sequence.

For anomaly detection, they first collect the information from tracking result including the number of objects, their motion history and interaction, the timing of their behaviors. Then they interpret events based on the basic information about WHO (how many), WHEN, WHERE and WHAT. Based on this interpretation, anomaly can be defined by some rules. For example, in a traffic intersection scenario, there are always 0 - 8 cars around. If at a time 15 cars arrive this intersection simultaneously, it can be regarded as an anomalous event.

C. Video Datasets and Intuitive Illustration

As discussed above, if an anomaly is generated by single object, we can call it single-object anomaly. Figs. 6 (a) and (b) show consecutive frames from 2 examples of single-object anomalies. In Fig. 6 (a), a man suddenly falls on the floor (see Fig. 6 (a2)) when walking across the lobby. In Fig. 6 (b), instead of turning left or right in front of the stop sign, the driver suddenly backs his car - see Figs. 6 (b3)-(b4). By setting the number of objects \( P = 1 \), our joint sparsity model reduces to the model by Li et al. [8] and can be used to detect anomalous trajectories of individual (single) objects.

On the other hand, if an anomaly happens via the interaction of multiple objects, we call it a multiple-object anomaly. The video clip frames from 3 examples of multiple-object anomalies are shown in Figs. 7 (a), (b) and (c). In Fig. 7 (a), a pedestrian walk crossing the street loses his hat and retraces...
his footsteps to pick it up from the road. At this time, a vehicle comes in very close proximity to the pedestrian and comes to a sudden halt - see Fig. 7(a2). In Fig. 7(b), the second vehicle (marked by a red rectangle) comes to a complete stop when waiting for the vehicle in front of it (Fig. 7(b1)), but does not actually stop at the stop sign - see Figs. 7(b2)-(b3). In Fig. 7(c), a car fails to yield to oncoming car while turning left - see Figs. 7(c2)-(c3). The examples in Fig. 7(a) and (b) are in fact from a real-world transportation database (which cannot be made public for proprietary reasons) which we refer to as the Xerox Stop Sign database. An example video clip is however made available at: [http://youtu.be/M6_Pjigg5CY](http://youtu.be/M6_Pjigg5CY) And the example in Fig. 7(c) comes from another proprietary transportation database which we address as the Xerox Intersection database. A representative video clip is made available at: [http://youtu.be/ZGKtkVtWEFU](http://youtu.be/ZGKtkVtWEFU).

Our proposed algorithm for multi-object trajectory based anomaly detection is called Joint Kernel-based Sparsity Model (abbreviated to JKSM) or equivalently Kernel Sparsity Model (KSM) in the case of single object anomaly detection. We test
the KSM and JKSM algorithms on several challenging video datasets. For single-object anomalies, we test on CAVIAR [29] data set in Fig. 6(a) and the Xerox Stop Sign video database - represented in Fig. 6(b). For multiple-object anomalies, we test on AVSS [30] data set - see Fig. 7(a), the Xerox Stop Sign data set - multi-object example in Fig. 7(b), and Xerox Intersection data set - see Fig. 7(c). We also compare our experimental results against three well-recognized techniques in trajectory based anomalous event detection: 1.) the recent sparsity based technique of Li et al. [18] using one class SVMs, 2.) the sparsity model with online dictionary learning of Zhao et al. [9] and 4.) the multiple-object tracking and rule-based anomaly detection technique of Han et al. [31].

Benefits of sparsity under object occlusion or missing trajectory information:

Because of the limitation of camera’s visual angle, occlusions often occur in video data. Figs. 8(a) and (b) show two examples of occlusions. In Fig. 8(a), a car is occluded by another car (we call it a moving occlusion)- see Fig. 8(a3). In Fig. 8(b), the car is occluded by the stop sign (static occlusion) - see Fig. 8(b2).

Tracking algorithms continue to strive to improve and perform well even under occlusion [43]–[45]. Since this is a fundamentally hard problem, occlusion occurs in practice and leads to missing or corrupted trajectory data. Because our optimization problems in (12) and (22) are well-conditioned, perturbation theory arguments apply and lend our method robustness under limited trajectory occlusion. In particular, consider:

\[
\hat{S}_o = \arg \min_{S_o} \| J( H \circ S_o) \|_{row,0} \quad \text{subject to} \quad \| Y_o - AS_o \|_F \leq \varepsilon, \tag{34}
\]

where \( Y_o \) is a set of occluded test trajectories and \( S_o \) denotes the corresponding sparse coefficients. If \( \| Y_o - Y \|_2 < \eta \) then \( \| \hat{S}_o - \hat{S} \|_2 < \zeta \), i.e. a small perturbation to the problem should cause only a small perturbation to the solution [46]. It means that if the occluded trajectory set \( Y_o \) is not very different from the original trajectory set \( Y \), then the optimized sparse coefficients under trajectory occlusion will be close enough to \( \hat{S} \), i.e. the solution in the absence of occlusion. Because anomaly detection rests on the structure of the sparse coefficient matrix, this intuitively provides occlusion robustness.

In addition, we perform a simple experiment to illustrate this. We work with the Xerox stop sign data set and 95 trajectories (including 76 training trajectories without occlusion and 19 occluded trajectories) obtained from 39 video clips. Our training dictionary consists of 9 normal trajectory classes (containing 8 trajectories each) and 1 anomalous trajectory classes (containing 4 trajectories). An independent set of 13 normal but occluded trajectories and 6 anomalous but occluded trajectories are used to test our approach. Fig. 9 shows an example of occluded trajectories from Xerox Stop Sign data. Note that occluded trajectory locations are replaced by a constant value (e.g. zero) for the duration that object tracking is lost. This is a common characteristic of frame-based tracking approaches [41], [47], [48].

Because training for anomalous events is well-represented in this database, Eq. (5) is used to classify the test trajectories. The RBF function is chosen as the kernel for the KSM algorithm. The confusion matrices of four methods - KSM and the techniques by Picarelli et al., Li et al. and Zhao et al. are reported in Table I. Note that all Li et al. and Zhao et al. which are also a sparsity based anomaly detection technique, as well the proposed KSM methods do better than Picarelli et al. owing to the robustness of sparse coefficients under occlusion. Further, KSM is mildly better than Li et al. and Zhao et al. for this example owing to the non-linearity in the sparsity model as introduced by the use of the kernel. A more thorough evaluation of the three methods on real-world databases is reported next.

D. Experimental Results

1) Detection Rates for Single-Object Anomaly Detection:

For the CAVIAR data set, we test on 27 video clips from which 170 trajectories are extracted. Our training dictionary consists of 10 normal trajectory classes and 3 anomalous trajectory classes, each class contains 10 different training trajectories. 21 normal trajectories and 19 anomalous trajectories are used as
TABLE I
CONFUSION MATRICES OF PROPOSED AND STATE-OF-THE-ART TRAJECTORY BASED METHODS ON XEROX STOP SIGN OCCLUDED DATA – SUPERVISED, SINGLE-OBJECT ANOMALY DETECTION, KSM DETECTS ANOMALIES USING Eq. (1)

<table>
<thead>
<tr>
<th>Method</th>
<th>Normal</th>
<th>Anomaly</th>
<th>Normal</th>
<th>Anomaly</th>
<th>Normal</th>
<th>Anomaly</th>
</tr>
</thead>
<tbody>
<tr>
<td>Li et al. [8]</td>
<td>61.5%</td>
<td>50.0%</td>
<td>92.6%</td>
<td>4.4%</td>
<td>76.9%</td>
<td>50.0%</td>
</tr>
<tr>
<td>Zhao et al. [9]</td>
<td>76.9%</td>
<td>23.1%</td>
<td>92.6%</td>
<td>4.4%</td>
<td>84.6%</td>
<td>61.5%</td>
</tr>
<tr>
<td>KSM</td>
<td>46.5%</td>
<td>46.5%</td>
<td>84.6%</td>
<td>15.4%</td>
<td>84.6%</td>
<td>15.4%</td>
</tr>
</tbody>
</table>

Fig. 9. An example of occluded trajectories from the Xerox Stop Sign video data.

Fig. 10. ROC curves for unsupervised single-object anomaly detection – CAVIAR video data set. KSM detects anomalies using Eq. (17), where $\tau_1$ varies from 0.1 to 0.9.

1 All training and test trajectories for both normal and anomalous trajectories are manually hand labeled by video analysis experts. The training and test sets picked for computing detection accuracy are completely non-overlapping.
For the AVSS data set, 3 different 2-object normal event classes (containing 24 training trajectory pairs each) are chosen. The database was experimentally found to contain 2 different anomalies - corresponding videos can be seen at: http://youtu.be/mU5R056zInc and http://youtu.be/EzLkWF65lo. Our outlier rejection measure in Eq. (18) was again successfully able to detect both these anomalies. Table V shows the detection rates of six methods.

In Xerox Intersection data, there are 91 trajectory pairs extracted from 13 video clips. We manually build our training dictionary into 6 different 2-object normal event classes (containing 6 trajectory pairs each) and 6 different anomalous classes(containing 4 trajectory pairs each). 17 normal trajectory pairs and 14 anomalous trajectory pairs are used for testing. Again, we use the RBF kernel function. The confusion matrices of our method against the three competing trajectory based techniques are shown in Table VI.

It is easy to see from Table VI that the proposed JKSM method leads to the best detection rates. The improvement over the techniques of Piciarelli et al. [18] is expected since this technique is really for single-object anomaly detection and the extensions Piciarelli et al. [18] is expected since this technique is really for single-object anomaly detection and the extensions Piciarelli et al. _1 and Piciarelli et al. _2 - will either strongly compromise detection or lead to high false alarm. For Zhao et al., spatio-temporal volumes are used as features, which can not distinguish between single-object and multiple-object cases. Therefore, the performance of Zhao et al. is slightly worse than Han et al., which is designed for multiple-object scenario. In [31], anomalies are detected by the use of context based rules on the result of multiple-object tracking. This puts an unreasonable burden on defining these rules and is often restrictive in practice, i.e. not all anomalies can be anticipated. In the proposed joint sparsity model, interactions between distinct object trajectories are better captured and departures from expected “joint behavior” (particularly in the

![Fig. 11. ROC curves for unsupervised single-object anomaly detection – Xerox Stop Sign video dataset. KSM detects anomalies using Eq. (17), where $\tau_1$ varies from 0.1 to 0.9.](image)

<table>
<thead>
<tr>
<th>TABLE II</th>
<th>Confusion Matrices of Proposed and State-of-the-Art Trajectory Based Methods on CAVIAR Data – Supervised, Single-Object Anomaly Detection. KSM Detects Anomalies using Eq. (3).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>Normal</td>
</tr>
<tr>
<td>Normal</td>
<td>85.7%</td>
</tr>
<tr>
<td>Anomaly</td>
<td>14.3%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>Normal</td>
</tr>
<tr>
<td>Normal</td>
<td>85.3%</td>
</tr>
<tr>
<td>Anomaly</td>
<td>14.7%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE IV</th>
<th>Confusion Matrices of Proposed and State-of-the-Art Trajectory Based Methods on the Xerox Stop Sign Data – Unsupervised Multiple-Object Anomaly Detection. The Proposed JKSM Detects Anomalies using Eq. (27) and a Threshold Value of 0.5.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Piciarelli <em>et al.</em> [18]</td>
</tr>
<tr>
<td>Detection Rates:</td>
<td>3/3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE V</th>
<th>Detection Rates of Proposed and State-of-the-Art Methods on AVSS Data – Unsupervised Multiple-Object Anomaly Detection. The Proposed JKSM Detects Anomalies using Eq. (27) and a Threshold Value of 0.5.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Piciarelli *et al._1</td>
</tr>
<tr>
<td>Detection Rates:</td>
<td>2/2</td>
</tr>
</tbody>
</table>

the joint event is anomalous. We denote this method as Piciarelli *et al._2*.

To ensure the application of Piciarelli *et al._1*, we separate every 2-object event into 2 individual events with known individual class labels (normal or anomalous) so that we can obtain results for the aforementioned extensions, i.e. Piciarelli *et al._1* and Piciarelli *et al._2*.

For the Xerox Stop Sign data set, we identify 4 different 2-object normal event classes. Each class contains 15 training trajectory pairs. This database inherently contains 3 multiple-object anomalies, one of which is illustrated in Fig. [7] (b) with the actual video at [http://youtu.be/PR-L3NmM](http://youtu.be/PR-L3NmM). Using the outlier rejection in Eq. (18), all three multiple-object anomalies are successfully detected by our proposed JKSM. The detection rates of all methods are shown in Table [IV].
TABLE VI
CONFUSION MATRICES OF PROPOSED AND STATE-OF-THE-ART TRAJECTORY BASED METHODS ON THE XEROX INTERSECTION DATASET – SUPERVISED, MULTIPLE-OBJECT ANOMALY DETECTION. THE PROPOSED JKSM DETECT ANOMALIES USING EQ. (26)

<table>
<thead>
<tr>
<th></th>
<th>Piciarelli et al. 1</th>
<th>Piciarelli et al. 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Normal</td>
<td>Anomaly</td>
</tr>
<tr>
<td>Normal</td>
<td>58.8%</td>
<td>7.1%</td>
</tr>
<tr>
<td>Anomaly</td>
<td>41.2%</td>
<td>92.9%</td>
</tr>
<tr>
<td></td>
<td>Normal</td>
<td>Anomaly</td>
</tr>
<tr>
<td>Normal</td>
<td>79.4%</td>
<td>35.7%</td>
</tr>
<tr>
<td>Anomaly</td>
<td>20.6%</td>
<td>64.3%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Zhao et al.</th>
<th>Han et al. [31]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Normal</td>
<td>Anomaly</td>
</tr>
<tr>
<td>Normal</td>
<td>79.4%</td>
<td>35.7%</td>
</tr>
<tr>
<td>Anomaly</td>
<td>20.6%</td>
<td>64.3%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>JKSM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Normal</td>
</tr>
<tr>
<td>Normal</td>
<td>88.2%</td>
</tr>
<tr>
<td>Anomaly</td>
<td>11.8%</td>
</tr>
</tbody>
</table>

Fig. 12. ROC curves for unsupervised multiple-object anomaly detection – Xerox Intersection dataset. The proposed JKSM detect anomalies using Eq. (27) where the detection threshold varies from 0.1 to 0.9.

case training for anomalies is absent/limited) is employed which enables the improvement in detection rates.

ROC Curves: By changing the threshold $\tau_2$ correspond to Eq. (18), $\theta_{th}$ in Eq. (30) and $\bar{\epsilon}$ in Eq. (32), we can generate the ROC curves for JKSM. (Jos a (joint sparsity model without the kernel), Piciarelli et al. and Zhao et al. (Here, we average the ROC curves of Piciarelli et al. 1 and Piciarelli et al. 2 as the ROC curve of Piciarelli et al.). For Han et al, we loosen and tighten the anomaly detection rules to obtain the relationship between true positive rate and false positive rate. The ROC curves of these four different methods on all 31 test trajectory pairs from Xerox Intersection data set are shown in Figs. 12. The statistical merits of the joint sparsity model are clearly revealed by the ROC curves. Again, the use of a kernel enables JKSM to further improve performance.

VI. CONCLUSION
We present a new joint sparsity model for video anomaly detection. An over-complete training dictionary is derived comprising joint observations of multi-object events; and a test event is reconstructed as a sparse linear combination of events in the training dictionary. For the joint sparsity model to be meaningful, constraints are posed on the structure of sparse coefficients which leads to a new simultaneous sparsity optimization problem. In the supervised case, where anomalies are pre-characterized into classes, the anomaly detection reduces to a sparsity based classification problem. In the more realistic unsupervised scenario where anomalies cannot be sufficiently pre-characterized, the anomaly detection is accomplished via a multi-object outlier rejection measure. The merits of a principled joint sparsity model for multi-object anomaly detection are strongly corroborated via experiments on challenging real-world databases. Additionally, we propose a kernelization of the joint sparsity model so as to further improve anomaly detection where linear sparse reconstruction models do not hold directly. While we use fixed, expert designed training dictionaries that be updated periodically, online dictionary updates/learning, e.g. in [9] is a useful extension of our work and can be pursued in future research.

REFERENCES


Xuan Mo received the Master’s degree in Automation from Tsinghua University, China, in 2010. He is currently a PhD candidate in the Electrical Engineering Department of the Pennsylvania State University. His research interests include computational color & imaging, signal processing and computer vision. His currently work focuses on video anomaly detection for transportation application.

Vishal Monga is currently an Assistant Professor of Electrical Engineering at the main campus of Pennsylvania State University in University Park, PA. He was with Xerox Research from 2005-2009. His undergraduate work was completed at the Indian Institute of Technology (IIT), Guwahati and his doctoral degree in Electrical Engineering was obtained from the University of Texas, Austin in Aug 2005. Dr. Mongas research interests are broadly in statistical signal and image processing. He established the Information Processing and Algorithms Laboratory (iPAL) at Penn State. Current research themes in his lab include computational color and imaging, multimedia mining and security, and robust image classification and recognition. He currently serves as an Associate Editor for the IEEE Transactions on Image Processing and the SPIE Journal of Electronic Imaging. While with Xerox Research in Rochester, he was selected as the 2007 Rochester Engineering Society (RES) Young Engineer of the Year. He is the recipient of a 2011 Air Force Faculty Fellowship and a Monkowski Early Career award from the college of engineering at Penn State.

Raja Bala received the Ph.D. degree in Electrical Engineering from Purdue University. He is currently a Principal Scientist with the Xerox Innovation Group. His research interests include color management, novel image rendering techniques, image processing for augmented reality applications, and image and video analytics. Dr. Bala holds over 150 patents and publications in the field of color and imaging. He is a Fellow of IS&T and member of IEEE.

Dr. Zhigang (Zeke) Fan received his MS and PhD degrees in electrical engineering from the University of Rhode Island, Kingston, RI, in 1986 and 1988, respectively. He joined Xerox Corporation in 1988 where he is currently a principal scientist in Xerox Corporate Research and Technology. Dr. Fans research interests include various aspects of image processing and recognition. He has authored and co-authored more than 90 technical papers, as well as over 200 patents and pending applications. Dr. Fan is a Fellow of SPIE and a Fellow of IS&T.