Online Thinning for High Volume Streaming Data*

Xin J. Hunt
SAS Institute Inc.
100 SAS Campus Dr
Cary, North Carolina 27513
xin.hunt@sas.com

Rebecca Willett
University of Wisconsin-Madison
Department of Electrical and Computer Engineering
Madison, Wisconsin 53706
willett@discovery.wisc.edu

ABSTRACT

In an era of ubiquitous large-scale streaming data, the availability of data far exceeds the capacity of expert human analysts. In many settings, such data is either discarded or stored unprocessed in data centers. This paper proposes a method of *online data thinning*, in which large-scale streaming datasets are winnowed to preserve unique, anomalous, or salient elements for timely expert analysis. At the heart of this proposed approach is an online anomaly detection method based on dynamic, low-rank Gaussian mixture models. Specifically, the high-dimensional covariance matrices associated with the Gaussian components are associated with low-rank models. According to this model, most observations lie near a union of subspaces. The low-rank modeling mitigates the curse of dimensionality associated with anomaly detection for high-dimensional data, and recent advances in subspace clustering and subspace tracking allow the proposed method to adapt to dynamic environments. The resulting algorithms are scalable, efficient, and are capable of operating in real time. Experiments on wide-area motion imagery illustrate the efficacy of the proposed approach.

KEYWORDS

subspace clustering, subspace tracking, online learning, Gaussian mixture model, anomaly detection, saliency detection

ACM Reference format:
Xin J. Hunt and Rebecca Willett. 2017. Online Thinning for High Volume Streaming Data. In *Proceedings of ACM SIGKDD Conference, El Halifax, Nova Scotia Canada, August 2017 (KDD'17),* 10 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Modern sensors are collecting high-dimensional data at unprecedented volume and speed; human analysts cannot keep pace. For instance, many sources of intelligence data must be translated by human experts before they can be widely accessible to analysts and actionable; the translation step is a significant bottleneck [55]. Typical NASA missions collect terabytes of data every day [28, 45, 48, 50]. Incredibly, the Large Hadron Collider (LHC) at CERN "generates so much data that scientists must discard the overwhelming majority of it—hoping hard they've not thrown away anything useful." [30] There is a pressing need to help analysts prioritize data *accurately and efficiently* from a storage medium or a data stream. This task is complicated by the fact that, typically, the data is neither thoroughly annotated nor meaningfully catalogued. Failure to extract relevant data could lead to incorrect conclusions in the analysis, while extraction of irrelevant data could overwhelm and frustrate human analysts, throttling the discovery process.

This paper focuses on *scalable online data processing algorithms that can winnow large datasets to produce smaller subsets of the most important or informative data for human analysts.* This process is described as "data thinning." Often, the data thinning process involves flagging observations which are inconsistent with previous observations from a specified class or category of interest, or are ranked highly according to a learned ranking function. Typically we are interested in methods which can perform these assessments from streaming data, as batch algorithms are inefficient on very large datasets.

One generic approach to the problem of data thinning for large quantities of (possibly streaming) high-dimensional data requires estimating and tracking a probability distribution f_t underlying the stream of observations x_t, and flagging an observation as anomalous whenever $\widehat{f}_t(x_t) < \tau$ for some small threshold $\tau > 0$, as demonstrated in past work [42, 57]. Ultimately, the goal is to ensure that the flagged data is salient to human analysts on the receiving end without being buried in an avalanche of irrelevant data. Within this general framework, there are three key challenges:

- **Dynamic environments:** The data may not be from a stationary distribution. For example, it may exhibit diurnal, location- or weather-dependent patterns. Effective data thinning methods must adapt to those dynamics and sources of bias. Global summary statistics and naive online learning algorithms will fail in this context.
- **High-dimensional data:** Individual data points x_t may be high-dimensional, resulting in the classical "curse of dimensionality" [13, 41]. While large quantities of data may be available, the combination of high-dimensional data and a non-stationary environment still results in an ill-posed estimation problem.
- **Real-time processing:** In applications like those with NASA and CERN, large quantities of streaming data preclude computationally intensive or batch processing.

1.1 Data thinning for wide-area motion imagery

While our approach is not restricted to imaging data, one important application of our data thinning approach is real-time video analysis. Recent advances in optical engineering have led to the advent of

*Produces the permission block, and copyright information

KDD'17, August 2017, El Halifax, Nova Scotia Canada

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

new imaging sensors that collect data at an unprecedented rate and scale; these data often cannot be transmitted efficiently or analyzed by humans due to their sheer volume. For example, the ARGUS system developed by BAE Systems is reported to collect video-rate gigapixel imagery [15, 38], and even higher data rates are anticipated soon [18, 20, 37]. This type of data is often referred to as wide-area motion imagery (WAMI). Currently WAMI streams are used primarily in a forensic context – after a significant event occurs (*e.g.*, a security breach), the data immediately preceding the event are analyzed *reactively* to piece together what led to that event. However, there is a strong need for predictive analysis which can be used to help *anticipate* or detect negative events in real time.

Unfortunately, the latter form of analysis is often infeasible for two reasons: (1) the data acquisition rate exceeds the capacity of many sensor platforms' downlinks; and (2) size, weight, and power constraints limit processing capabilities on airborne sensor platforms. Thus an *emerging and fundamental challenge is efficiently downloading salient information to ground-based analysts over a limited-bandwidth channel.* While data compression has a long history, conventional compression methods may distort information particularly relevant to analysts. In particular, standard motion imagery compression techniques typically focus on optimizing peak signal-to-noise ratio or psycho-visual metrics which apply globally to an entire video and are often unrelated to any specific task.

Instead, a better solution would be to identify unique objects or regions of WAMI, and transmit only features of these objects. This concept is illustrated in Fig. 1. Ideally, this method will identify regions and features of a data stream most critical to a given task, and prioritize these features when preparing data for storage or transmission. This task is clearly related to "visual saliency detection" (*cf.,* [40, 43, 44, 58]); we describe the connections between the proposed work and saliency detection in Section 2.

Note that in this setting a key challenge is that the sensor may be placed on a vibrating platform that introduces significant jitter into the data and precludes direct comparison of successive frames. While real-time video stabilization has been considered in the video processing literature (*cf.,* [11, 25, 35, 39, 59]), such methods are often robust for small motions associated with a hand-held device and break down with large motions associated with mechanical vibrations. More robust methods capable of processing larger degrees of jitter can be computationally prohibitive on energy-constrained platforms.

1.2 Problem formulation and approach

Suppose we are given a sequence of data $x_1, x_2, \ldots$, and for $t = 1, 2, \ldots, x_t \in \mathbb{R}^p$, where p denotes the *ambient dimension.* Assume that x_t comes from some unknown distribution, *i.e.,* there exists some sequence of distributions P_t such that

$$x_t \sim P_t \quad t = 1, 2, \ldots$$

where P_t evolves over time, and its distribution density function is denoted by f_t. The goal is to find the x_t that are unusual or anomalous. In particular, we assign each observation x_t an *anomalousness score* proportional to its negative log likelihood under the estimated model—*i.e.,* $-\log f_t(x_t)$. Observations with a high anomalousness score can then either be directed to a human analyst or flagged for further processing and analysis.

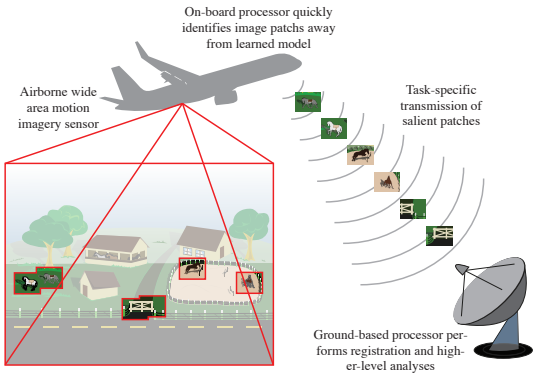


Figure 1: Conceptual illustration of proposed objectives. An airborne platform collects wide-area motion imagery (WAMI), identifies task-specific salient patches, and transmits only those patches. The ground-based receiver can then perform more sophisticated processing, including registration, geolocation, and activity analysis.

The key challenge here is two-fold: (a) the dimension of the signal, p, can be quite large, and (b) f_t may evolve rapidly over time. The combination of these factors means that our problem is ill-posed.

This paper proposes a method for estimating and tracking the time-series of density functions f_t over $\mathbb{R}^p$. In stationary, low-dimensional settings, we might consider a Gaussian mixture model that could be estimated, for instance, using an online expectation-maximization (EM) algorithm [61]. However, the non-stationary setting and high dimensions make that approach unviable, as we demonstrate experimentally later in the paper. The proposed approach, by contrast, considers a constrained class of Gaussian mixture models in which the Gaussian covariance matrices (each in the positive-semidefinite cone $\mathcal{S}_+^p$) are low-rank. This model is equivalent to assuming most x_t lie near a union of low-dimensional subspaces. While this union of subspaces is unknown *a priori*, we may leverage recent advances in subspace tracking (*cf.,* [10, 27, 49]) and subspace clustering (*cf.,* [1–3, 5, 16, 19]) to yield an accurate sequence of density estimates $\hat{f}_t$, and mitigate the curse of dimensionality.

2 RELATED WORK

While data thinning is an emerging concept associated with modern high-dimensional, high-velocity data streams, the formulation described in Section 1.2 is closely related to anomaly detection, visual saliency detection, and subspace clustering and tracking.

2.1 Anomaly detection

The study of anomaly detection has a long and rich history [32]. However, most existing detection methods do not work well with high dimensional data, and often do not work online. A 2009 survey [24] categorizes the available methods into six different categories, of which cluster-based methods, statistical anomaly detection methods, and spectral methods are most related to this work.

Cluster-based methods (*cf.,* [23, 56]) work by first assigning data into clusters, then compute anomaly score based on the data's cluster assignment. The computational costs of these methods are usually high, and the performance highly depends on the distance measure, which are often problem-dependent.

Certain statistical methods (*cf.,* [4, 6]) assume that the data are drawn from some standard or predetermined distribution, and determines outliers by computing the likelihood of the signal coming from such distributions. These methods do not rely on a big training set, but estimating the distribution of high-dimensional data is a non-trivial task, and the statistical assumptions do not always hold true.

Spectral methods (*cf.,* [7, 31]) assume that data can be embedded into a lower dimensional subspace, and detect anomalies over the embedded space rather than the original space.Spectral methods are well-suited to high-dimensional data. However, they can incur high computational costs; even online anomaly detection algorithms (*cf.,* [8, 46]) face this challenge. Furthermore, the subspace model underlying spectral methods is less flexible than the union of subspace model underlying this paper's proposed method.

2.2 Visual saliency detection

In the special case of imagery or video data, data thinning is closely related to visual saliency detection. Like anomaly detection, saliency detection has been widely studied over the last few decades. A standard benchmark for comparison in image saliency detection is proposed by Itti et al. in [44]. This paper attempts to explain human visual search strategies, using biologically motivated algorithms. However, this algorithm is too slow to apply to real time videos. Hou and Zhang in [43] use spectral analysis to detect salient objects for faster speed. However, the analysis breaks down when multiple types of salient objects are present in the scene. Graph-based methods (*cf.,* [40]) has very good performance, but suffers from high computational complexity. The cluster-based algorithm proposed in [58] works better than [44], but not as well as the graph-based algorithms. The information theoretic model based algorithm proposed in [22] works as well as [44], but requires much less tuning. [72] improved the work of [22], with better performance and faster speed.

Methods for image saliency detection have been extended to video saliency detection, but those methods assume a stable imaging platform and video stream free of jitter. In the WAMI application described above, however, sensors can be placed on vibrating platforms that preclude most video saliency detection methods.

2.3 Subspace clustering and tracking

The proposed method is also closely related to the subspace clustering and tracking algorithms. Subspace clustering methods cluster observations into low-dimensional subspaces to mitigate the curse of dimensionality, which often make nearest-neighbors-based methods inaccurate [14]. Correlation clustering methods (*cf.,* [1–3, 5, 16, 19]) can identify multiple arbitrarily angled subspaces at the same time, but all share the same problem of high computational cost. Even [1], which is shown to beat other methods in speed, still has an overall complexity of $O(p^2T^2)$, where p is the dimension of the problem, and T is the total number of data points. More recent methods based on sparse modeling (*cf.,* [33, 34, 54, 65, 66]) require solving convex optimization problems that can be inefficient in high-dimensional settings. Thus, the high complexity of the algorithms make them less than ideal candidates for an efficient online algorithm.

Subspace tracking is a classical problem that experienced recent attention with the development of algorithms that are robust to missing and outlier elements of the data points x_t. For example, the Grassmannian Rank-One Update Subspace Estimation (GROUSE) [10], Parallel Estimation and Tracking by REcursive Least Squares (PETRELS) [27], and Robust Online Subspace Estimation and Tracking Algorithm (ROSETA) [49] effectively track a single subspace using incomplete data vectors. These algorithms are capable of tracking and adapting to changing environments. The subspace model used in these methods, however, is inherently strong, whereas a plethora of empirical studies have demonstrated that high-dimensional data often lie near manifolds with non-negligible curvature [9, 12, 60].

In contrast, the non-parametric mixture of factor analyzers [26] uses a mixture of low-dimensional approximations to fit to unknown and spatially-varying (but static) curvatures. The Multiscale Online Union of SubSpaces Estimation (MOUSSE) method developed by Xie et al. [70] employs union of subspaces tracking for change point detection in high-dimensional streaming data. Thanks to the adoption of the state-of-the-art subspace tracking techniques, the algorithm is both accurate and efficient (with complexity linear in p). However, MOUSSE cannot be directly applied for our data thinning task for a few reasons. First, MOUSSE is designed for change-point detection and does not have a probabilistic model. Thus observations in a rare subspace would still be treated as typical, which makes it difficult to discover the rare observations. Second, MOUSSE can only process one observation at a time, *i.e.,* it does not allow for mini-batch updates that can be helpful in data thinning applications, where data could arrive in blocks. Last but not least, although MOUSSE is able to deal with missing data, [70] does not explore the computational-statistical tradeoffs that are important for time- or power-sensitive applications. This paper presents a method that is designed for the data thinning task, has a specific statistical model, and allows for mini-batch updates which increases the algorithm's efficiency.

3 DATA THINNING VIA TRACKING UNION OF SUBSPACES

3.1 Union of subspaces model

Recall from Section 1.2 that each $x_t \in \mathbb{R}^p$ is assumed to be drawn from a distribution with density f_t, and that f_t is modeled as a mixture of Gaussians where each Gaussian's covariance matrix is the sum of a rank-r matrix (for $r < p$) and a scaled identity matrix. We refer to this as a *dynamic low-rank GMM*. In particular, the j^{th} Gaussian mixture component is modeled as

$$\mathcal{N}\left(\mu_{j,t}, \Sigma_{j,t}\right)$$

where $\mu_{j,t} \in \mathbb{R}^p$ is the mean and $\Sigma_{j,t} = V_{j,t}\Lambda_{j,t}V_{j,t}^T + \sigma_j^2 I$.

Here $V_{j,t} \in \mathbb{R}^{p\times r}$ is assumed to have orthonormal columns, and $\Lambda_{j,t} \in \mathbb{R}^{r\times r}$ is a diagonal matrix with positive diagonal entries. If $\sigma_j = 0$, then $\Sigma_{j,t}$ would be rank-r and any point drawn from that

Gaussian would lie within the subspace spanned by the columns of $V_{j,t}$ – shifted by $\mu_{j,t}$. By allowing $\sigma_j > 0$ we model points drawn from this Gaussian lying near that r-dimensional shifted subspace. Overall, we model

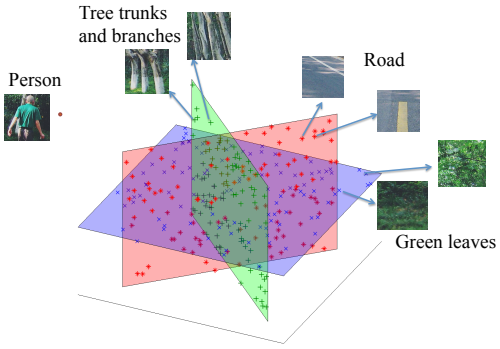
$$f_t = \sum_{j=1}^{K_t} q_{j,t} \mathcal{N}\left(\mu_{j,t}, V_{j,t}\Lambda_{j,t}V_{j,t}^T + \sigma_j^2 I\right) \quad (1)$$

where K_t is the number of mixture components in the model at time t and $q_{j,t}$ is the probability of x_t coming from mixture component j.

To better understand this model, we can think of each observation x_t as having the form $v_t + w_t$, where v_t lies in a union of subspaces (or more precisely, because of the Gaussian means, a union of affine subspaces) defined by the $V_{j,t}$s and within ellipsoids embedded in those subspaces, where the ellipsoid axis lengths are determined by the $\Lambda_{j,t}$s.

Fig. 2 illustrates the union of subspaces model. Fig. 2a shows a sample image where one person is walking on a road with trees on both sides [63]. We would want to learn from a sequence of such images that the trees, grass and the road which occupy most of the pixels are typical of the background, and label the person as salient because it is uncommon in the scene. Fig. 2b illustrates the union of subspaces model. When we divide the image into patches, the vast majority of patches are plant and road patches, and only a few patches contain the person. The plant and road patches live on a union of subspaces as illustrated and can be thinned, leaving anomalous patches for further analysis.

(a) Image of a pedestrian walking on a road with trees on the sides



(b) Illustration of the union of subspaces idea

Figure 2: Illustration of the union of subspaces idea. Fig. 2a shows a pedestrian walking on a road with trees on the sides [63]. The road and the plants occupy most of the pixels, and they can be considered living in a union of subspaces. The person on the road would be considered as an outlier.

3.2 Algorithm highlights

This section explains how the proposed method estimates the evolving Gaussian mixture model using the techniques from the union of subspaces tracking algorithms. These steps are summarized in in Fig. 3. As seen, this data thinning method shares some features with the online EM algorithm for GMM estimation. However, there are a few key differences which are elaborated below:

- We constrain covariances to lie in a *union of subspaces*, which significantly reduces the problem size for estimating the covariance matrices. This constraint improves the accuracy of the algorithm, and also makes our method much stabler when the environment is changing rapidly relative to the data availability. This constraint also reduces computation time.
- In some settings, such as when working with WAMI data, we receive groups of x_t's simultaneously and can perform model updates more efficiently using *mini-batch techniques* (more details in Section 3.3.3).
- For large, high-velocity data streams, real-time processing is paramount. Even evaluating the likelihood of each new observation can be time consuming. The subspace model and mini-batch approach we take reduces computation complexity.
- For the online GMM estimation algorithm [61], the number of mixture components is selected *a priori*, and does not change for the duration of the task. Existing method like [36] can estimate the number of mixture components, but the algorithm is batch-based and does not easily adapt to streaming data. The proposed method *adapts to changing numbers of mixture components*, which allows the mixture model to better track the environmental dynamics. The method adapts the number of mixture components using a multiscale representation of a hierarchy of subspaces. (More details about the multi-scale model is discussed in Section 3.3.)

3.3 The Online Thinning algorithm

This section describes the updates of the parameters associated with the proposed dynamic low-rank GMM in (1). The updates of the mixture component weights ($q_{j,t}$) and means ($\mu_{j,t}$) are computed using stochastic gradient descent. The updates of the covariance matrices are more sophisticated and leverage subspace tracking methods.

The biggest challenge is updating K_t, the number of mixture components. In real-life applications, the number of mixture components is in general (a) not known *a priori*, and (b) can change with t. Thus a mechanism for adaptively choosing the number of subspaces is needed. Reducing model order is slightly less challenging because

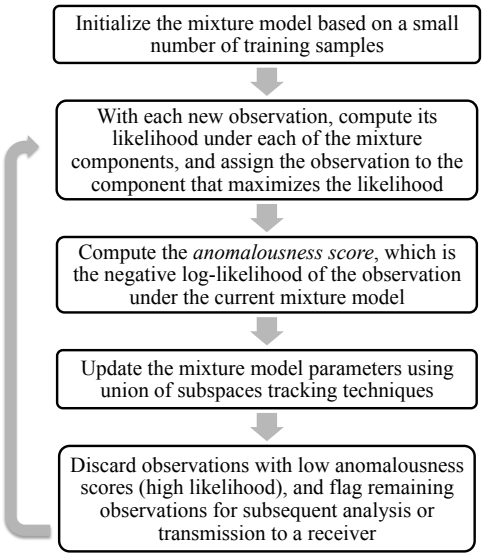


Figure 3: Flow chart of the main steps in the data thinning method.

it is relatively simple to merge two nearby mixture components. However, increasing model order is a much more complex issue, especially in an online setting.

To address these challenges, we organize these mixture components using a tree structure, as illustrated in Fig. 4. The idea for a multiscale tree structure stems from the multiscale harmonic analysis literature [29] and online updates of such models are introduced in [70]. In our setting, at time t, the j^{th} node is associated with a Gaussian distribution parameterized by its mean vector $\mu_{j,t}$, low-rank covariance matrix parameters $V_{j,t}, \Lambda_{j,t}$, and weight $q_{j,t}$. Most of the probability mass associated with each Gaussian is an ellipsoid centered at $\mu_{j,t}$, where $V_{j,t}$ and $\Lambda_{j,t}$ characterize the principle axes and principal axis lengths, respectively, of the ellipsoid. Finally, $q_{j,t}$ is approximately the probability of an observation falling inside this ellipsoid.

In the tree structure, we denote the set of leaf nodes as $\mathcal{J}_t \triangleq \{j : j^{\text{th}}$ node is a leaf node at time $t\}$ and have $K_t \triangleq |\mathcal{J}_t|$. The leaves of the tree correspond to the Gaussian mixture components in the model shown in Eq. (1). Each parent node corresponds to a single Gaussian which approximates the weighted sum of the Gaussians associated with its two children, where the weights correspond to the children's q parameters. Each of the tree leaves is also associated with two *virtual* children nodes. The virtual children nodes correspond to their own Gaussian distributions that can be used to grow the tree. The decision of pruning and growing are made based on (a) the accuracy of the Gaussian mixture model, *i.e.,* the cumulative (with a forgetting factor) anomalousness score, and (b) the size of the mixture model, *i.e.,* the total number of leaf nodes at time t.

3.3.1 Computation of the Gaussian mixture likelihood (and anomalousness score). The proposed algorithm uses the negative log-likelihood of the Gaussian mixture model give the data point as its anomalousness score. The likelihood of x_t under the Gaussian associated with node j is given by (recall $\Sigma_{j,t} = V_{j,t}\Lambda_{j,t}V_{j,t}^T + \sigma_j^2 I$)

$$p_{j,t}(x_t) = \frac{1}{(2\pi)^{p/2}|\Sigma_{j,t}|^{1/2}} e^{-\frac{1}{2}(x_t-\mu_{j,t})^T\Sigma_{j,t}^{-1}(x_t-\mu_{j,t})}. \quad (2)$$

Using the model in Eq. (1), the Gaussian mixture negative log-likelihood function (and hence anomalousness score) for any $x_t \in \mathbb{R}^p$ is:

$$s_t(x_t) = -\log f_t(x_t) = -\log\left(\sum_{j\in\mathcal{J}_t} q_{j,t}p_{j,t}(x_t)\right). \quad (3)$$

3.3.2 Selective update. With the observation of each x_t, the algorithm first compute the likelihood of x_t under each of the Gaussian mixture components, and then assign x_t to the component that maximizes the likelihood. Specifically, after the likelihood computations above, x_t is assigned to the mixture component $j_t^* \triangleq \arg\max_{j\in\mathcal{J}_t}\{p_{j,t}(x_t)\}$. Note that the weights $q_{j,t}$ are not used here in order to avoid biasing towards components with large weights. This assignment is made in order to reduce the computational complexity of the parameter update step: with each x_t, instead of updating all the parameters of the entire tree, the algorithm only updates the tree branch associated leaf node j_t^*. That is, the algorithm updates the parameters of node j_t^*, all of its ancestors, and one of node j_t^*'s virtual children (the one under which x_t is more likely). This approach significantly reduces the time complexity of the updates, especially when the model is complex (*i.e.,* when the number of leaf nodes is large).

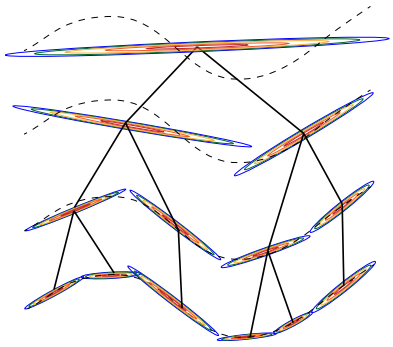


Figure 4: Multiscale representation of low-rank Gaussian mixture model. Consider a density with its mass concentrated along the black dashed curve. Each successive level in the multiscale representation has more Gaussian mixture components (depicted via contour plots) with covariance matrices corresponding to more compact ellipsoids, and hence yields a more accurate approximation of the underlying density. Given a particular binary tree representation of a GMM, the approximation error can be allowed to increase or decrease by pruning or growing the binary tree connecting the different scales. The ellipsoids are all very compact along some axes because they correspond to covariance matrices that are the sum of a low-rank matrix and a scaled identity matrix.

3.3.3 Mini-batch update. In many applications, multiple observations can arrive simultaneously. For example, in WAMI settings, hundreds of image patches in a single frame arrive at the same time. One way to deal with this is simply treat each patch as arriving at a different time, and update the model parameters separately with each observation. However, when the number of patches is large (*e.g.,* for HD videos, there can be thousands of patches per frame), this sequential processing can be extremely time-consuming.

To reduce the computation cost, we instead update the mixture model in mini-batches. When multiple observations are received at the same time, we first compute the anomalousness score of each observation, and assign them to their own mixture component. The collection of observations assigned to a given mixture component then form a mini-batch. The tree nodes and tree structure are then updated once per mini-batch. When the size of mini-batches is much larger than 1, this approach significantly reduces the number of times needed to update the mixture component parameters and tree structures. Note that this mini-batch processing does not affect the computation of the anomalousness score and component assignment, where each observation is processed sequentially.

Thus, now we assume that we receive a collection of observations stored in matrix $X_t = [x_{t,1}, \dots, x_{t,N_t}] \in \mathbb{R}^{p \times N_t}$ at time t, where $x_{t,i} \in \mathbb{R}^p$ for all $i = 1, \dots, N_t$. A special case of this is $N_t = 1$, which is the sequential update without mini-batches. After assigning each column in X_t to the K_t leaf nodes in the hierarchical tree structure based on their distance to the corresponding mixture components, we can rewrite X_t into mini-batches, $X_t = [X_{j_1,t}, \cdots, X_{j_{K_t},t}]$, where $\{j_1, \dots, j_{K_t}\} \subseteq \mathcal{J}_t$. Here each $X_{j_i,t} \in \mathbb{R}^{p \times n_{j_i,t}}, i = 1, \dots, K_t$ is a block of $n_{j,t}$ data points that are assigned to the j_i^{th} node in the tree (must be a leaf node). Note that $\sum_{j \in \mathcal{J}_t} n_{j,t} = N_t$.

Our update equations are based on a "forgetting factor" $\alpha \in (0, 1)$ that places more weight on more recent observations; this quantity affects how quickly a changing distribution is tracked and is considered a tuning parameter to be set by the end user. Then for each leaf node j that needs updates (*i.e.,* with assigned observations), the weights $q_{j,t}$ are then updated by

$$q_{j,t+1} = \alpha q_{j,t} + (1-\alpha)\frac{n_{j,t}}{N_t}. \tag{4}$$

Note that for the leaf nodes the weights need to add to 1, *i.e.,* $\sum_{j \in \mathcal{J}_t} q_{j,t} = 1$ for all t. If we initialize $q_{j,1}$ s.t. $\sum_{i \in \mathcal{J}_1} q_{j,1} = 1$, and the weight of any parent node is the sum of the weights of its two children, then this update preserves $\sum_{i \in \mathcal{J}_t} q_{j,t} = 1$ for all t. The mixture component means $\mu_{j,t}$ are updated by

$$\mu_{j,t+1} = \alpha \mu_{j,t} + \frac{(1-\alpha)}{n_{j,t}} X_{j,t} \mathbb{1}_{n_{j,t} \times 1}. \tag{5}$$

The diagonal matrix $\Lambda_{j,t} \triangleq \text{diag}\{\lambda_{j,t}^{(1)}, \dots, \lambda_{j,t}^{(r)}\} \in \mathbb{R}^{r \times r}$, with $\lambda_{j,t}^{(1)}, \dots, \lambda_{j,t}^{(r)} \geq 0$, contains eigenvalues of the covariance matrix of the projected data onto each subspace. Let

$$M_{j,t} = [\mu_{j,t}, \dots, \mu_{j,t}] \in \mathbb{R}^{p \times n_{j,t}} \tag{6}$$

be a means matrix computed by concatenating $n_{j,t}$ copies of $\mu_{j,t}$ together. Let

$$B_{j,t} = V_{j,t}^{\#}(X_{j,t} - M_{j,t}) \tag{7}$$

be the residual signal, where the superscript $^{\#}$ denotes the pseudo-inverse of a matrix (for orthonormal $V_{j,t}$, the pseudo-inverse is its transpose). Denote its m^{th} row as $B_{j,t}^{(m)}$. Then we can update

$$\lambda_{j,t+1}^{(m)} = \alpha \lambda_{j,t}^{(m)} + (1-\alpha)\|B_{j,t}^{(m)}\|_2^2, m = 1, \dots, r. \tag{8}$$

The subspace matrices $V_{j,t}$ are updated using Algorithm. 1. The updates of $V_{j,t}$ and $\Lambda_{j,t}$ are a mini-batch extension of the PETRELS [27] update equations, with an added step of orthonormalization of $V_{j,t+1}$ since PETRELS does not guarantee the orthogonality of $V_{j,t+1}$.

For the ancestors of each leaf node that need updates, we combine all the mini-batches assigned to its children, and update the node with the same formulae as above using the combined mini-batches. For the virtual children of leaf nodes that need updates, we divide each mini-batch into two sub-mini-batches based on the likelihood of each observation under the Gaussian of the virtual node, and update each virtual node with its assigned sub-mini-batch.

Algorithm 1 Mini-Batch Update of Covariance Parameters

1: **Initialize:** $V_{j,1}$ (with training data), $R_{j,1} = c\mathbb{1}_{r \times r}, c \ll 1$
2: **input:** $X_{j,t}, V_{j,t}, R_{j,t}, M_{j,t}$
3: $B_{j,t} = V_{j,t}^{\#}(X_{j,t} - M_{j,t})$
4: $R_{j,t+1} = \alpha R_{j,t} + B_{j,t}B_{j,t}^T$
5: $\widetilde{V}_{j,t+1} = V_{j,t} + \left((X_{j,t} - M_{j,t})B_{j,t}^T - V_{j,t}B_{j,t}B_{j,t}^T\right) R_{j,t+1}^{\#}$
6: **Orthonormalization**
$V_{j,t+1} = \widetilde{V}_{j,t+1}\left(\widetilde{V}_{j,t+1}^T \widetilde{V}_{j,t+1}\right)^{-\frac{1}{2}}$
7: **Output:** $V_{j,t+1}, R_{j,t+1}$

3.3.4 Tree structure update. The growing (splitting nodes) and pruning (merging nodes) of the tree structure allow the complexity of the GMM to adapt to the diversity of the observed data. The number of nodes in the tree controls the tradeoff between the model accuracy and complexity. The proposed method determines whether to grow or prune the tree by greedily minimizing a cost function consisting of the weighted cumulative anomalousness score (with weights corresponding to the forgetting factor α described above) and the model complexity ($|\mathcal{J}_t|$).

Define ϵ_t as the cumulative anomalousness score where $\epsilon_0 = 0$, and $\epsilon_{t+1} = \alpha\epsilon_t + \frac{1}{N_t}\sum_{i=1}^{N_t} s_t(x_{t,i})$. For each node j (including virtual children), a similar cumulative score $e_{j,t}$ is kept based only on the mini-batches assigned to that node. Let $\mathcal{I}_{j,t} \triangleq \{i : x_{t,i}$ assigned to j^{th} node or its children$\}$ (for virtual nodes this set is the indices of its sub-mini-batch), initialize $e_{j,0} = 0$, and $e_{j,t}$ is updated by

$$e_{j,t+1} = \alpha e_{j,t} + \frac{1}{|\mathcal{I}_{j,t}|}\sum_{i \in \mathcal{I}_{j,t}} -\log\left(p_{j,t}(x_{t,i})\right). \tag{9}$$

Let TOL be a pre-set error tolerance. For each leaf node $j_1 \in \mathcal{J}_t$ that is assigned new observations, let j_0 be its parent, j_2 be its sibling, and $j_{1,1}, j_{1,2}$ be its virtual children. Let γ be a positive constant. Split node j_1 if

$$\epsilon_{t+1} \leq \text{TOL}, \tag{10}$$

and

$$e_{j_1,t} + \gamma K_t > \frac{q_{j_{1,1},t} e_{j_{1,1},t} + q_{j_{1,2},t} e_{j_{1,2},t}}{q_{j_{1,1},t} + q_{j_{1,2},t}} + \gamma(K_t + 1). \tag{11}$$

Note the left side of Ineq. (11) is the penalized cumulative score of node j_1 (where the penalty is proportional to the number of nodes in the tree), while the right side of Eq. (11) is the average penalized cumulative score of node j_1's two virtual children. We split node j_1 if the average penalized cumulative score is smaller at the virtual children level.

Similarly, merge nodes j_1 and j_2 if

$$\epsilon_{t+1} \geq \text{TOL} \tag{12}$$

and

$$e_{j_0,t} + \gamma(K_t - 1) < \frac{q_{j_1,t} e_{j_1,t} + q_{j_2,t} e_{j_2,t}}{q_{j_1,t} + q_{j_2,t}} + \gamma K_t, \tag{13}$$

Note the left side of Ineq. (13) is the penalized (with tree size) cumulative score of node j_1's parent j_0, while the right side of Eq. (11) is the average penalized cumulative score of node j_1 and its sibling j_2. We merge j_1 and j_2 if the average penalized cumulative score of j_1 and j_1 is larger than the penalized score of their parent. The use of these penalized scores to choose a tree which is both (a) a good fit to the observed data and (b) as small as possible to avoid overfitting is common in classification and regression trees [21, 53, 62, 67–69]. The splitting and merging operations are detailed in Algorithm 2 and Algorithm 3. The complete Online Thinning algorithm is summarized in Algorithm 4.

Algorithm 2 Grow tree

Input: Node j with virtual children nodes k and ℓ
Update $\mathcal{J}_{t+1} = \mathcal{J}_t \bigcup \{k, \ell\} \backslash \{j\}$
Create new virtual children: k_1, k_2 for new leaf node k, and ℓ_1, ℓ_2 for new leaf node ℓ
Let $v_{i,t}^{(1)}$ be the first column of $V_{i,t}, i \in \{k, \ell\}$
Initialize virtual nodes k_1, k_2, ℓ_1 and ℓ_2:
  for $i \in \{k, \ell\}$

$$\mu_{i_1,t+1} = \mu_{i,t} + \sqrt{\lambda_{i,t}^{(1)}} v_{i,t}^{(1)}/2, \quad \mu_{i_2,t+1} = \mu_{i,t} - \sqrt{\lambda_{i,t}^{(1)}} v_{i,t}^{(1)}/2$$
$$V_{i_1,t+1} = V_{i,t}, \quad V_{i_2,t+1} = V_{i,t}$$
$$\lambda_{i_1,t+1}^{(1)} = \lambda_{i,t}^{(1)}/2, \quad \lambda_{i_2,t+1}^{(1)} = \lambda_{i,t}^{(1)}/2$$
$$\lambda_{i_1,t+1}^{(m)} = \lambda_{j,t}^{(m)}, \quad \lambda_{i_2,t+1}^{(m)} = \lambda_{j,t}^{(m)}, \quad m = 2, \ldots, r$$
$$q_{i_1,t+1} = q_{j,t}/2, \quad q_{i_2,t+1} = q_{j,t}/2$$

Algorithm 3 Prune tree

Input: Node j with children nodes j_1 and j_2 to be merged
Delete all four virtual children nodes of j_1 and j_2
Update $\mathcal{J}_{t+1} = \mathcal{J}_t \bigcup \{j\} \backslash \{j_1, j_2\}$
Define j_1, j_2 as the virtual children nodes of the new leaf node j

Algorithm 4 Online Thinning with Mini-Batch Updates

Input: error tolerance TOL > 0, threshold $\tau > 0$, forgetting factor $\alpha \in (0, 1)$
Initialize: tree structure, set initial error $\epsilon_1 = 0$
for $t = 1, 2, \ldots$ **do**
  Receive new data $X_t \in \mathbb{R}^{p \times N_t}$
  for $i = 1, 2, \ldots, N_t$ **do**
    Let $x_{t,i}$ be the i^{th} column of X_t
    For all $j \in \mathcal{J}_t$, compute likelihood using Eq. (2)
    Compute anomalousness score $s_t(x_{t,i})$ using Eq. (3)
    Assign $x_{t,i}$ to leaf $j_t^* \triangleq \arg\max_{j \in \mathcal{J}_t} \{p_{j,t}(x_{t,i})\}$
    Compute the likelihood of $x_{t,i}$ under j^*'s two virtual children nodes, and also assign $x_{t,i}$ to the virtual child with higher likelihood
  end for
  Update $\epsilon_{t+1} = \alpha\epsilon_t + \frac{1}{N_t} \sum_{i=1}^{N_t} s_t(x_{t,i})$
  for all nodes j in the tree **do**
    $\mathcal{I}_{j,t} \triangleq \{i : x_{t,i}$ assigned to j^{th} node or its children$\}$
    if $\mathcal{I}_{j,t}$ is not empty **then**
      Denote all data assigned to node j or its children as $X_{j,t} = [x_1, \ldots, x_{n_{j,t}}]$
      Compute (9), (4), (5), (6), (7), (8)
      Update $V_{j,t}$ by calling Algorithm 1
      if Ineq. (10) and (11) **then** call Algorithm 2
      else if Ineq. (12) and (13) **then** call Algorithm 3
      end if
    else update $q_{j,t+1} = \alpha q_{j,t}$
    end if
  end for
  $\mathcal{X}_t = \{x_{t,i} : s_t(x_{t,i}) > \tau\}$
end for
Output: sequence of thinned data $\mathcal{X}_1, \ldots, \mathcal{X}_T$

4 SYNTHETIC DATA EXPERIMENTS

This section compares the Online Thinning approach based on tracking a dynamic low-rank GMM with (a) a classical full-rank batch (static) GMM [51, 52], (b) a full-rank online GMM estimation algorithm [61], and (c) MMLE-MFA [71], a low-rank batch (static) GMM estimation algorithm. The classical batch GMM and online GMM algorithms do not have the low-rank structure exploited by the Online Thinning algorithm and MMLE-MFA. While MMLE-MFA incorporates the low-rank model, the algorithm is batch-based and does not evolve over time.

For these experiments, the data is generated as follows: The ambient dimension is $p = 100$. We first generate points in $\mathbb{R}^p$ in a union of three (shifted) subspaces of dimension ten; in which 95% of the points lie in the union of the first two subspaces. The other 5% of the points lie in a third subspace that is orthogonal to the other two. All three subspaces have shifts close to 0. We then add white Gaussian noise with standard deviation $\sigma = 0.01$ to these points to generate our observations. The two subspaces where the 95% of observations come from are dynamic, where the subspaces rotate at a speed $\delta \geq 0$. For $j = 1, 2$, we have $V_{j,t+1} = V_{j,t} + \delta \frac{B}{\|B\|_F} V_{j,t}$, where B is a $p \times p$ skew-symmetric matrix. Denote the set of x_t's coming

from each of the three subspaces as $\mathcal{X}_j, j = 1, 2, 3$, respectively. The goal is to identify the 5% of the observations that come from $\mathcal{X}_3$.

The experiment streams in nine thousand observations in total. For Online Thinning and the full-rank online GMM, an initial model is estimated using the first one thousand samples, and the models are then updated in an online fashion for the remaining eight thousand samples. For the full-rank batch GMM algorithm and the MMLE-MFA algorithm, we estimate a GMM model on the entire nine thousand data points (after all samples come in) at once. The anomalousness score is calculated as the negative log-likelihood of each data point according to the estimated model.

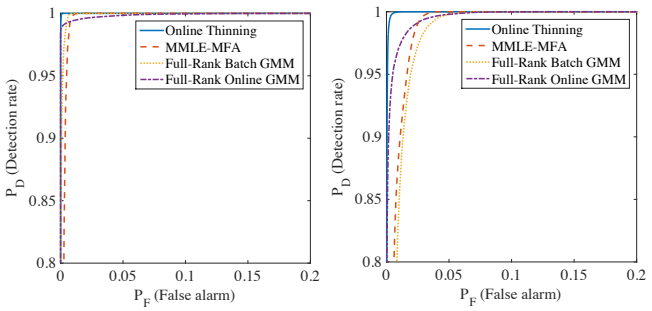
Fig. 5 compares the detection accuracy (in ROC curves) of Online Thinning and the two comparator algorithms in two settings, where in 5a, the true subspaces used to generate the data are kept static ($\delta = 0$) throughout the experiment, and in 5b, the true subspaces rotate at the rate of $\delta = 2 \times 10^{-2}$ at each time step. Each plotted experiment is averaged over one hundred random realizations. As seen in the plots, all four algorithms perform well when the mixture components are static, while Online Thinning outperforms the the other three algorithms when the subspaces change over time.

The average time used by the four algorithms to process the nine thousand samples is (a) Online Thinning: 2.70s, (b) full-rank batch GMM: 4.18s, (c) full-rank online GMM: 16.38s, and (d) MMLE-MFA: 62.64s.[1] Online Thinning is significantly faster than the other three algorithms. The computation time is recorded with single-threaded implementations in MATLAB (2015a) with a 2.2GHz CPU.

The performance gap can be explained by the underlying models of the four algorithms. Both the full-rank batch GMM and full-rank online GMM algorithms rely on full-rank GMM models, which make estimating the covariance matrices difficult without a large number of observations. Furthermore, the MMLE-MFA and the batch full-rank GMM algorithm rely on static models, which introduce bias when the environment is dynamic. On the other hand, Online Thinning is based on a dynamic low-rank GMM model, and thus faces a much less ill-posed problem by having a union of subspace assumption (which significantly reduces the number of unknowns in the covariance matrices). Also, Online Thinning focuses on the most recent samples by weighing down the past samples, and can quickly adapt to the changes in the subspaces.

In this experiment, the Online Thinning algorithm also correctly estimates the number of mixture components as two, while for the other three algorithms, the correct number of mixture components is given *a priori*. To further demonstrate the ability to estimate the number of mixture components by Online Thinning, we generate data with varying number of mixture components, and record the estimated number of components by the Online Thinning algorithm. In this experiment, we generate three segments with 10^4 samples each ($p = 100$). For the first and third segment, the non-anomalous data (95% of samples) come from two different rank-ten mixture components, while in the second segment, the non-anomalous data comes from four different rank-ten mixture components. The mixture components stay static within each segment. Fig. 6 shows the number of mixture components estimated by Online Thinning

[1]MMLE-MFA is originally developed for compressive sensing where our problem can be seen as a special case where the sensing matrices are identity matrices. We modified the code for MMLE-MFA to reduce computation time involved with the identity sensing matrices.



(a) Static subspaces **(b) Subspaces changing at rate 2×10^{-2}**

Figure 5: Comparison between Online Thinning using a dynamic low-rank GMM, a static low-rank GMM (MMLE-MFA), a classical online GMM, and a classical static batch GMM.

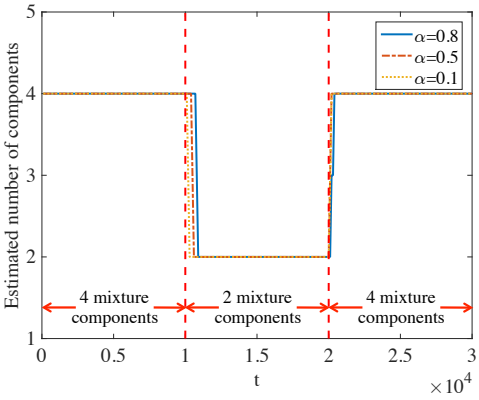
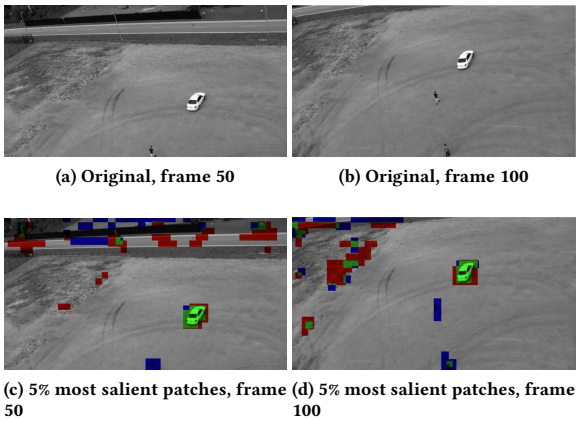


Figure 6: Number of mixture components estimated by Online Thinning with different α values. For the first and third 10^4 samples, the non-anomalous data (95% of samples) comes from four different rank-ten components, while in the second 10^4 samples, the non-anomalous data comes from two different rank-ten components.

with three different α values. The vertical red dash lines indicate when the number of non-anomalous mixture components change. As seen, for all three values of α, the Online Learning algorithm quickly adapts to the correct number of components after both changes. The smaller α is, the faster the algorithms adapts after a sudden change.

5 WAMI EXPERIMENTS

This experiment compares Online Thinning with the SUN (Saliency Using Natural statistics) algorithm proposed by Zhang *et al* in [72]. We do not use the other GMM-based algorithms discussed in Section 4 due to (a) the MMLE-MFA and full-rank batch GMM are not online algorithms, which means they are unsuitable for streaming video applications, and (b) though full-rank online GMM is an online algorithm, the high computational complexity does not allow for video processing in realistic settings. The SUN algorithm, on the



(a) Original, frame 50 (b) Original, frame 100

(c) 5% most salient patches, frame 50 (d) 5% most salient patches, frame 100

Figure 7: Data thinning result using Online Thinning and SUN algorithms on the surveillance video at frames 50 and 100. The first row shows the original video, and the second row shows the data thinning results. In the results, green patches are flagged by both methods, blue patches are only flagged by Online Thinning, and red patches are only flagged by SUN.

other hand, is representative of the state-of-the-art visual saliency detection algorithms [17], provides a general framework for many models, performs as well as or better than previous models, and is computationally efficient [72]. Thus the SUN algorithm is a better algorithm for real-life video applications.

We perform this comparison on a real surveillance video capturing an empty field near a highway. In the video, a car is parked on the lot, and two people can be seen walking in and out of the scene on the field. We use this video because it is clear that the car and the people are most salient in the scene. The original video can be found at https://youtu.be/mX1TtGdGFMU. For the Online Thinning algorithm, we use SIFT (scale-invariant feature transform) features [47] of frame t as our observation X_t at time t. Specifically, we use the package from [64] to compute the dense SIFT features (*i.e.,* SIFT features computed over a pre-set grid of points on each frame) as features. Each frame of the video is of size 960×540, and the grid is placed so that one SIFT feature is computed for each 25×25 patch. Each frame have roughly eight hundred SIFT feature vectors. The dimension of each SIFT feature vector is 128.

Fig. 7 shows the result of Online Thinning and the SUN algorithms on this surveillance video at frames 50 and 100. Figures 7a and 7b show the original frames, while in 7c and 7d, we flag the top 5% patches with the highest anomalousness or saliency scores by the Online Thinning and SUN algorithms. In the results, green patches are flagged by both methods, blue patches are only flagged by Online Thinning, and red patches are only flagged by SUN. Note that in both frames, the people in the scene are mostly labeled by blue, *i.e.,* they are only flagged by Online Thinning. The Online Thinning outperforms the SUN algorithm by more consistently flagging small rare patches such as the people; this is in part due to the adaptivity of Online Thinning to dynamic environments. The result video can be found at https://www.youtube.com/watch?v=DyLJThawgi0.

6 CONCLUSION

This paper proposed an online data thinning method for high-dimensional data with changing environment. At the heart of the proposed algorithm is a union of subspaces tracking algorithm, which allows for fast and accurate data thinning in a variety of applications.

The core idea of the proposed approach is to track a Gaussian mixture model whose covariance matrices each are dominated by a low-rank component. Under this model, most observations are concentrated in a union of subspaces, a model growing in popularity in image, video, and text analysis because of its flexibility and robustness to over-fittings. Unlike traditional GMMs, the low-rank structure proposed here mitigates the curse of dimensionality and facilitates efficient tracking in dynamic environments. Furthermore, by leveraging the recent advances in subspace tracking and subspace clustering techniques, the proposed method is able to accurately estimate the mixture density without adding a significant computational burden. Another important feature of the proposed method is the ability to track an arbitrary number of mixture components. The adoption of a tree-like hierarchical structure for the union of subspaces model allows the method to adaptively choose the number of subspaces needed at each time stamp, and thus greatly improves the flexibility of the method and accuracy when tracking highly dynamic densities.

ACKNOWLEDGMENTS

This work is supported by 14-NGI-1084 and IIS-1447449.

REFERENCES

[1] E. Achtert, C. Böhm, H.-P. Kriegel, P. Kröger, and A. Zimek. 2007. On exploring complex relationships of correlation clusters. In *Scientific and Statistical Database Management, 2007. SSBDM'07. 19th International Conference on.* IEEE, 7–7.
[2] E. Achtert, C. Böhm, H.-P. Kriegel, P. Kröger, and A. Zimek. 2007. Robust, Complete, and Efficient Correlation Clustering.. In *SDM.* SIAM, 413–418.
[3] E. Achtert, C. Böhm, P. Kröger, and A. Zimek. 2006. Mining hierarchies of correlation clusters. In *Scientific and Statistical Database Management, 2006. 18th International Conference on.* IEEE, 119–128.
[4] D. Agarwal. 2007. Detecting anomalies in cross-classified streams: a bayesian approach. *Knowledge and information systems* 11, 1 (2007), 29–44.
[5] C. C. Aggarwal and P. S. Yu. 2000. *Finding generalized projected clusters in high dimensional spaces.* Vol. 29. ACM.
[6] C. C. Aggarwal and P. S. Yu. 2008. Outlier Detection with Uncertain Data.. In *SDM*, Vol. 483. SIAM, 493.
[7] A. Agovic, A. Banerjee, A. R. Ganguly, and V. Protopopescu. 2008. Anomaly Detection in Transportation Corridors Using Manifold Embedding. *Knowledge Discovery from Sensor Data* (2008), 81–105.
[8] T. Ahmed. 2009. Online anomaly detection using KDE. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE.* IEEE, 1–8.
[9] W. K. Allard, G. Chen, and M. Maggioni. 2012. Multi-scale geometric methods for data sets II: Geometric multi-resolution analysis. *Applied and Computational Harmonic Analysis* 32, 3 (2012), 435–462.
[10] L. Balzano, R. Nowak, and B. Recht. 2010. Online identification and tracking of subspaces from highly incomplete information. In *Proc. Allerton Conf. on Comm., Control and Comp.* 704 – 711.
[11] S. Battiato, A. R. Bruna, and G. Puglisi. 2010. A robust block-based image/video registration approach for mobile imaging devices. *Multimedia, IEEE Transactions on* 12, 7 (2010), 622–635.
[12] M. Belkin and P. Niyogi. 2003. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation* 15, 6 (2003), 1373–1396.
[13] R. Bellman. 1961. *Adaptive control processes: a guided tour.* Vol. 4. Princeton University Press, Princeton, NJ.
[14] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. 1999. When is "nearest neighbor" meaningful? In *Database Theory ICDT'99.* Springer, 217–235.
[15] D. Bezier. 2007. BAE to Develop Surveillance System. *The Washington Post* (2007). Retrieved 3-20-2012.

[16] C. Böhm, K. Kailing, P. Kröger, and A. Zimek. 2004. Computing clusters of correlation connected objects. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*. ACM, 455–466.

[17] A. Borji and L. Itti. 2013. State-of-the-art in visual attention modeling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 35, 1 (2013), 185–207.

[18] K. Bourzac. 2012. Gigapixel camera catches the smallest details. *Nature News* (2012).

[19] P. S. Bradley and O. L. Mangasarian. 2000. k-Plane clustering. *Journal of Global Optimization* 16, 1 (2000), 23–32.

[20] D. J. Brady, M. E. Gehm, R. A. Stack, D. L. Marks, D. S. Kittle, D. R. Golish, E. M. Vera, and S. D. Feller. 2012. Multiscale gigapixel photography. *Nature* 486 (2012), 386–389.

[21] L. Breiman, J. Friedman, R. Olshen, and C. J. Stone. 1983. *Classification and Regression Trees*. Wadsworth, Belmont, CA.

[22] N. Bruce and J. Tsotsos. 2005. Saliency based on information maximization. In *Advances in neural information processing systems*. 155–162.

[23] S. Budalakoti, A. Srivastava, R. Akella, and E. Turkov. 2006. Anomaly detection in large sets of high-dimensional symbol sequences. *NASA Ames Research Center, Tech. Rep. NASA TM-2006-214553* (2006).

[24] V. Chandola, A. Banerjee, and V. Kumar. 2009. Anomaly detection: A survey. *ACM computing surveys (CSUR)* 41, 3 (2009), 15.

[25] H.-C. Chang, S.-H. Lai, and K.-R. Lu. 2006. A robust real-time video stabilization algorithm. *Journal of Visual Communication and Image Representation* 17, 3 (2006), 659–673.

[26] M. Chen, J. Silva, J. Paisley, C. Wang, D. Dunson, and L. Carin. 2010. Compressive sensing on manifolds using a non-parametric mixture of factor analyzers: algorithm and performance bounds. *IEEE Trans. on Signal Processing* 58, 12 (2010), 6140 – 6155.

[27] Y. Chi, Y. C. Eldar, and R. Calderbank. 2012. PETRELS: Subspace estimation and tracking from partial observations. In *IEEE Int. Conf. on Acoustics, Speech and Sig. Proc. (ICASSP)*.

[28] W. Clavin. 2013. Managing the Deluge of 'Big Data' From Space. (2013). http://www.jpl.nasa.gov/news/news.php?release=2013-299.

[29] D. Donoho. 1997. CART and best-ortho-basis selection: A connection. *Annals of Stat.* 25 (1997), 1870–1911.

[30] E. Dumbill. 2012. What is big data? An introduction to the big data landscape. (2012). http://strata.oreilly.com/2012/01/what-is-big-data.html.

[31] H. Dutta, C. Giannella, K. D. Borne, and H. Kargupta. 2007. Distributed Top-K Outlier Detection from Astronomy Catalogs using the DEMAC System.. In *SDM*. SIAM, 473–478.

[32] F. Y. Edgeworth. 1887. Xli. on discordant observations. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 23, 143 (1887), 364–375.

[33] E. Elhamifar and R. Vidal. 2009. Sparse subspace clustering. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2790–2797.

[34] E. Elhamifar and R. Vidal. 2013. Sparse subspace clustering: Algorithm, theory, and applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 35, 11 (2013), 2765–2781.

[35] S. Ertürk. 2002. Real-time digital image stabilization using Kalman filters. *Real-Time Imaging* 8, 4 (2002), 317–328.

[36] M. A. T. Figueiredo and A. K. Jain. 2002. Unsupervised learning of finite mixture models. *IEEE Transactions on pattern analysis and machine intelligence* 24, 3 (2002), 381–396.

[37] L. Greenemeier. 2011. Draw the Curtains: Gigapixel Cameras Create Highly Revealing Snapshots. (2011). http://www.scientificamerican.com/article.cfm?id=gigapixel-camera-revealed.

[38] D. Hambling. 2009. Special ForcesÕ Gigapixel Flying Spy Sees All. (2009). http://www.wired.com/dangerroom/2009/02/gigapixel-flyin/.

[39] M. Hansen, P. Anandan, K. Dana, G. Van der Wal, and P. Burt. 1994. Real-time scene stabilization and mosaic construction. In *Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on*. IEEE, 54–62.

[40] J. Harel, C. Koch, and P. Perona. 2006. Graph-based visual saliency. In *Advances in neural information processing systems*. 545–552.

[41] T. Hastie, R. Tibshirani, and J. Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag New York. https://doi.org/doi:10.1007/978-0-387-84858-7

[42] C. Horn and R. Willett. 2011. Online anomaly detection with expert system feedback in social networks. In *Proc. International Conference on Acoustics, Speech, and Signal Processing*.

[43] X. Hou and L. Zhang. 2007. Saliency detection: A spectral residual approach. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 1–8.

[44] L. Itti, C. Koch, and E. Niebur. 1998. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 20, 11 (1998), 1254–1259.

[45] Z. Ivezic and et al. (108 additional authors not shown). 2008. LSST: from Science Drivers to Reference Design and Anticipated Data Products. (2008). http://arxiv.org/abs/0805.2366arXiv:0805.2366.

[46] H.-P. Kriegel, A. Zimek, et al. 2008. Angle-based outlier detection in high-dimensional data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 444–452.

[47] D. G. Lowe. 1999. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, Vol. 2. Ieee, 1150–1157.

[48] LSST Team. 2014. LSST: A New Telescope Concept. (2014). http://www.lsst.org/lsst/public/tour_software, retrieved June 7, 2014.

[49] H. Mansour and X. Jiang. 2015. A Robust Online Subspace Estimation and Tracking Algorithm. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. http://www.merl.com/publications/TR2015-027

[50] J. Marlow. 2012. What to Do With 1,000,000,000,000,000,000 Bytes of Astronomical Data per Day. (2012). Wired, http://www.wired.com/2012/04/what-to-do-with-1000000000000000000-bytes-of-astronomical-data-per-day/.

[51] G. McLachlan and T. Krishnan. 2007. *The EM algorithm and extensions*. Vol. 382. John Wiley & Sons.

[52] V. Melnykov, R. Maitra, et al. 2010. Finite mixture models and model-based clustering. *Statistics Surveys* 4 (2010), 80–116.

[53] R. Nowak, U. Mitra, and R. Willett. 2004. Estimating Inhomogeneous Fields Using Wireless Sensor Networks. *IEEE Journal on Selected Areas in Communications* 22, 6 (2004), 999–1006.

[54] D. Pimentel-Alarcon, L. Balzano, R. Marcia, R. Nowak, and R. Willett. 2016. Group-sparse subspace clustering with missing data. In *Proc. Statistical Signal Processing Workshop*.

[55] W. Pincus and D. Priest. June 20, 2002. NSA Intercepts On Eve of 9/11 Sent a Warning: Messages Translated After Attacks. (June 20, 2002). The Washington Post, Page A01.

[56] A. Pires and C. Santos-Pereira. 2005. Using clustering and robust estimators to detect outliers in multivariate data. In *Proceedings of the International Conference on Robust Statistics*.

[57] M. Raginsky, R. Willett, C. Horn, J. Silva, and R. Marcia. 2012. Sequential anomaly detection in the presence of noise and limited feedback. *IEEE Trans. Info. Theory* 58, 8 (Aug. 2012), 5544 – 5562.

[58] N. Rao, J. Harrison, T. Karrels, R. Nowak, and T. T. Rogers. 2010. Using machines to improve human saliency detection. In *Signals, Systems and Computers (ASILOMAR), 2010 Conference Record of the Forty Fourth Asilomar Conference on*. IEEE, 80–84.

[59] K. Ratakonda. 1998. Real-time digital video stabilization for multi-media applications. In *Circuits and Systems, 1998. ISCAS'98. Proceedings of the 1998 IEEE International Symposium on*, Vol. 4. IEEE, 69–72.

[60] S. T. Roweis and L. K. Saul. 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science* 290, 5500 (2000), 2323–2326.

[61] A. Samé, C. Ambroise, and G. Govaert. 2007. An online classification EM algorithm based on the mixture model. *Statistics and Computing* 17, 3 (2007), 209–218.

[62] C. Scott, R. D. Nowak, et al. 2006. Minimax-optimal classification with dyadic decision trees. *IEEE transactions on information theory* 52, 4 (2006), 1335–1353.

[63] Unsplash. 2015. (2015). https://pixabay.com/en/street-alley-trees-lined-road-828889/.

[64] A. Vedaldi and B. Fulkerson. 2008. VLFeat: An Open and Portable Library of Computer Vision Algorithms. http://www.vlfeat.org/. (2008).

[65] R. Vidal. 2010. A tutorial on subspace clustering. *IEEE Signal Processing Magazine* 28, 2 (2010), 52–68.

[66] Y.-X. Wang and H. Xu. 2016. Noisy Sparse Subspace Clustering. *Journal of Machine Learning Research* 17, 12 (2016), 1–41.

[67] R. Willett and R. Nowak. 2003. Platelets: a multiscale approach for recovering edges and surfaces in photon-limited medical imaging. *IEEE Transactions Med. Imaging* 22, 3 (2003), 332–350.

[68] R. Willett and R. Nowak. 2005. Level set estimation via trees. In *Proc. of ICASSP*.

[69] R. Willett and R. Nowak. 2007. Multiscale Poisson intensity and density estimation. *IEEE Transactions Information Theory* 53, 9 (2007), 3171–3187.

[70] Y. Xie, J. Huang, and R. Willett. 2013. Change-point detection for high-dimensional time series with missing data. *Selected Topics in Signal Processing, IEEE Journal of* 7, 1 (2013), 12–27.

[71] J. Yang, X. Liao, X. Yuan, P. Llull, D. J. Brady, G. Sapiro, and L. Carin. 2015. Compressive sensing by learning a Gaussian mixture model from measurements. *IEEE Transactions on Image Processing* 24, 1 (2015), 106–119.

[72] L. Zhang, M. H. Tong, T. K. Marks, H. Shan, and G. W. Cottrell. 2008. SUN: A Bayesian framework for saliency using natural statistics. *Journal of vision* 8, 7 (2008), 32.