MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Robust Low Rank Dynamic Mode Decomposition for Compressed Domain Crowd and Traffic Flow Analysis

Dicle, C.; Mansour, H.; Tian, D.; Benosman, M.; Vetro, A.

TR2016-087 July 2016

Abstract

In this paper, we develop a dynamic mode decomposition algorithm that is robust to both inlier and outlier noise in the data. One application of our algorithm is the identification of multiple crowd or traffic flows from compressed video streams. Our method uses motion vectors that are readily available in the compressed bitstream, and do not require computationally expensive optical flow. These motion vectors are known to be very noisy, however, our algorithm is able to extract the underlying dynamical systems that define the flows. We formulate a rank regularized dynamic mode decomposition problem with total least squares constraints to estimate the Koopman modes of the motion dynamics. The estimated Koopman modes are then used to analyze the stability of the system and extract steady state and transient flows. We demonstrate the improved performance of our approach compared to state of the art schemes and illustrate it applicability in identifying transient and steady-state flows in real video sequences.

IEEE International Conference on Multimedia and Expo (ICME)

ROBUST LOW RANK DYNAMIC MODE DECOMPOSITION FOR COMPRESSED DOMAIN CROWD AND TRAFFIC FLOW ANALYSIS

Caglayan Dicle[1*], *Hassan Mansour*[2], *Dong Tian*[2], *Mouhacine Benosman*[2], *Anthony Vetro*[2]

[1] Northeastern University, Boston, MA
[2] Mitsubishi Electric Research Laboratories, Cambridge, MA
dicle.c@husky.neu.edu, {mansour, tian, benosman, avetro}@merl.com

ABSTRACT

In this paper, we develop a dynamic mode decomposition algorithm that is robust to both inlier and outlier noise in the data. One application of our algorithm is the identification of multiple crowd or traffic flows from compressed video streams. Our method uses motion vectors that are readily available in the compressed bitstream, and do not require computationally expensive optical flow. These motion vectors are known to be very noisy, however, our algorithm is able to extract the underlying dynamical systems that define the flows. We formulate a rank regularized dynamic mode decomposition problem with total least squares constraints to estimate the Koopman modes of the motion dynamics. The estimated Koopman modes are then used to analyze the stability of the system and extract steady state and transient flows. We demonstrate the improved performance of our approach compared to state of the art schemes and illustrate it applicability in identifying transient and steady-state flows in real video sequences.

Index Terms— Dynamic mode decomposition, crowd analysis, Koopman modes, low rank matrix recovery, total least squares.

1. INTRODUCTION

The identification of motion flows within dense crowds of people in surveillance video sequences is an essential tool in crowd safety and crowd control tasks. Surveillance video of crowded scenes exhibit complex crowd behaviors even under normal situations. For example, crowd flows in large congested train stations can appear chaotic. However, it is often the case that low dimensional dynamical structures exist in the observed flow. Such structures are desirable to identify and segment from unstructured and transient flows. The automatic identification of different types of crowd flows aids in the monitoring and prediction of hazardous situations in crowded environments. Similarly, the identification of anomalous flows in traffic scenes helps management facilities predict and react to potential congestion.

Of particular interest is the detection and estimation of flows using motion information extracted from video streams. Considering individuals in a crowd scene or cars in a traffic scene as particles in a flow, the motion vectors of a single video frame correspond to observations of the velocities of particles at a time instance in the flow. Motion vectors can be computed using optical flow estimation applied to the video texture, or they can be parsed directly from the compressed domain of encoded video bitstreams. However, the extracted motion vectors are generally noisy and do not reflect the true motion of the particles. This is especially true in the case of compressed domain motion vectors which can include many outliers due to the rate-distortion optimization employed by most modern video encoders.

There as been extensive research in the field of crowd analysis. We briefly review some related techniques in Section 2. The dynamics of a crowd can be modeled at the micro (local) and macro (global) scales as the motion of particles in a fluid flow [1]. We consider in this paper the global scale and review as an example the Hughes model in Section 3 which characterizes the flow using a partial differential equation (PDE) in terms of the flow density and velocity. Identifying the system dynamics requires finding solutions to the PDE given initial and boundary conditions. Alternatively, dynamic mode decomposition (DMD) has recently been proposed in the fluid dynamics community as a data-driven and equation-free method for identifying system dynamics [2, 3]. Given time dependent observations of a linear or nonlinear dynamical system, DMD computes a temporal prediction matrix using an Arnoldi-like algorithm that estimates the infinite dimensional Koopman operator [4]. The success of DMD has recently seen broad use in multidimensional time series analysis including fluid modeling and video foreground background separation [5, 6]. However, the Arnoldi approach remains sensitive to noise in the observations. Several techniques have been developed to overcome the effect of noise in the data including sparsity promoting DMD (sp-DMD) [7] and low rank DMD with total least squares constraints (tls-DMD) [8]. We provide further details on DMD and some of the techniques that robustly compute the DMD operator in Section 4.

In this paper, we propose a **f**actorized **lo**w **ra**nk DMD algorithm (flora-DMD) that computes low rank factors of the DMD operator while satisfying total least squares constraints on the noise and outliers. The proposed approach, described in Section 5, scales well with large problems and addresses the same problem as tls-DMD with additional robustness to outliers in the data. We use an alternating direction method of multiplier (ADMM) framework for solving the rank regularized total least squares problem and demonstrate in Section 6 that our optimization formulation achieves better robustness to noise on synthetic examples than existing DMD schemes. Finally, we illustrate the applicability of our scheme for identifying steady state and transient flows in crowd and traffic video sequences.

2. RELATED WORK ON CROWD ANALYSIS

Early works on crowd analysis rely on particle advection in the optical flow field. [9] applies particle advection to extract Lagrangian Coherent Structures (LCS). Solmaz et. al. [10] uses LCS to find interest points in the motion field, and then characterizes the flow using an eigenvalue analysis of the motion Jacobian around the interest points. The work in [11–14] focus on individual or small group trajectories and can be considered complementary to our approach.

[*] This work was conducted while Caglayan was a intern at MERL.

Other works, such as the present work and [15], [16] focus on understanding the dominant flow components in the scene. [15] models the spatio-temporal behavior of the flow using mixture of Gaussian based clustering followed by an extrapolation based on conditional expectations. [16] assumes a strong affinity in the motion field and similarly constructs a probabilistic framework which can be manipulated using Lie algebra. Our method is less restrictive since it does not impose any geometric restrictions on the scene.

Another closely related branch of work is dynamic texture analysis, especially methods from [17] and [18]. Similar to [17], [18] we model the flow field as a dynamical system with a very high dimensional output. The advantage of our approach is that we do not limit our model to linear systems. We use the linear approximation operator DMD of the Koopman operator, which can also capture nonlinear dynamics in the data [2].

3. DYNAMICAL SYSTEM MODELING

When the density of a crowd is high, the motion of individuals in the crowd can be modeled as a fluid flow. One commonly used model for crowd analysis is the Hughes model [19] that models a crowd flow as a function of its density $\rho(x, y, t)$ and velocity $(\mathbf{u}(x, y, t), \mathbf{v}(x, y, t))$ as follows

$$\frac{\partial}{\partial t}\rho + \frac{\partial}{\partial x}(\rho \cdot \mathbf{u}) + \frac{\partial}{\partial y}(\rho \cdot \mathbf{v}) = 0, \tag{1}$$

where $\mathbf{u}(x, y, t)$ and $\mathbf{v}(x, y, t)$ are the respective velocities in the horizontal and vertical directions of every point (x, y) and time t.

The solution to (1) results in the crowd density map ρ and consequently the velocity fields $(\mathbf{u}, \mathbf{v})$ for all (x, y, t) that satisfy given initial and boundary conditions. Although, the dimensionality of the differential equations governing the evolution of ρ and $(\mathbf{u}, \mathbf{v})$ is theoretically infinite, it is often the case that the flows exhibit low dimensional behavior. Denote by $\mathbf{x}(t)$ the low dimensional state variable at time t, for which an observable vector $\mathbf{y}(t) = G(\mathbf{x}(t))$ corresponds to stacking of the density and velocity fields for all positions x and y at time t. The function G is a mapping from the low finite dimensional manifold $\mathcal{X}$ on which $\mathbf{x}$ evolves to the space of observables. Then, the solution to (1) projected onto $\mathcal{X}$ determines the transient response and stability of the corresponding dynamical system generally characterized by

$$\dot{\mathbf{x}}(t) = F(\mathbf{x}(t)), \tag{2}$$

where $F(\cdot)$ is some mapping in the low dimensional manifold $\mathcal{X}$ on which the dynamical system evolves. If (1) is further descretized in time, then the dynamical system evolution is characterized by

$$\mathbf{x}_{k+1} = F(\mathbf{x}_k), \tag{3}$$

where k is a time index.

4. KOOPMAN AND DYNAMIC MODE DECOMPOSITION

4.1. The Koopman operator

The Koopman operator [2, 4] is a linear operator K that satisfies the equation

$$\begin{aligned} G(F(\mathbf{x}_k)) &= KG(\mathbf{x}_k) \\ \Leftrightarrow \quad \mathbf{y}_{k+1} &= K\mathbf{y}_k. \end{aligned} \tag{4}$$

Although the dynamical system is nonlinear and evolves on a finite dimensional manifold, the Koopman operator is linear but infinite dimensional. Spectral analysis of the Koopman operator can be used to decompose the flow in terms of Koopman modes and associated Koopman eigenvalues that determine the temporal behavior of the corresponding Koopman mode.

4.2. Dynamic mode decomposition

Estimating the Koopman modes can be achieved through dynamic mode decomposition (DMD). Consider the data snapshot matrices

$$\mathbf{Y}_1 = \begin{bmatrix} | & & | \\ \mathbf{y}_0 & \dots & \mathbf{y}_{m-1} \\ | & & | \end{bmatrix}; \mathbf{Y}_2 = \begin{bmatrix} | & & | \\ \mathbf{y}_1 & \dots & \mathbf{y}_m \\ | & & | \end{bmatrix}. \tag{5}$$

The dynamic mode decomposition finds the best fit matrix $\mathbf{K}$ that satisfies the relation

$$\mathbf{Y}_2 \approx \mathbf{K}\mathbf{Y}_1. \tag{6}$$

Moreover, the eigenvectors and eigenvalues of $\mathbf{K}$ approximate the Koopman modes and Koopman eigenvalues. In what follows we shall use the terms Koopman modes and DMD modes interchangeably.

Let $\mathbf{u}_k$ and $\mathbf{v}_k$ be the horizontal and vertical motion vectors of all spatial blocks in a video frame indexed by k. We construct the observation $\mathbf{y}_k = [\mathbf{u}_k^T, \mathbf{v}_k^T]^T$ by vertically stacking the horizontal and vertical motion vectors in a vector $\mathbf{y}_k$. The time evolution of the motion vectors can now be modeled using an operator $\mathbf{K}$ by fitting the data to the one step prediction model

$$\mathbf{y}_k = \mathbf{K}\mathbf{y}_{k-1}. \tag{7}$$

Notice that (7) ensures that the same operator $\mathbf{K}$ models the evolution of both $\mathbf{u}_k$ and $\mathbf{v}_k$.

Next, suppose that we have $m+1$ observations of motion vectors from $m + 1$ video frames. We can then compute the operator $\mathbf{K}$ by forming the data matrices $\mathbf{Y}_1$ and $\mathbf{Y}_2$ similar to (5) and finding the operator $\mathbf{K}_{\text{ls}}$ that achieves the least squares fit, i.e.,

$$\begin{aligned} \mathbf{K}_{\text{ls}} &= \arg\min_K \tfrac{1}{2}\|\mathbf{Y}_2 - K\mathbf{Y}_1\|_F^2 \\ &= \mathbf{Y}_2\mathbf{Y}_1^\dagger, \end{aligned} \tag{8}$$

where the superscript † indicates the Moore-Penrose pseudo-inverse of a matrix, and $\|\cdot\|_F$ is the Frobenius norm of a matrix.

If the motion vectors correspond to the true motion of pedestrians in the video sequence, then the operator $\mathbf{K}_{\text{ls}}$ captures the full dynamics of the flow in the system. However, motion vectors, especially those extracted from an encoded sequence, are generally very noisy and often contain motion vectors that do not correspond to the true motion but are specified by the rate-distortion optimization engine employed in the video encoder.

4.3. Sparsity promoting DMD and total least squares DMD

Sparsity promoting DMD (sp-DMD) [7] is a two-step algorithm where in the first step all the modes are computed from the data, and in the second step a sparse subset of those modes are selected to explain the data. The first step performs a rank-r singular value decomposition (SVD) of the data matrix $\mathbf{Y}_1 = \mathbf{U}\Sigma\mathbf{V}^T$ to compute the reduced dimension DMD matrix

$$\begin{aligned} \mathbf{J} &:= \mathbf{U}^T\mathbf{Y}_2\mathbf{V}\Sigma^{-1} \\ &= \mathbf{\Psi}\mathbf{D}_\mu\mathbf{\Psi}^\dagger, \end{aligned} \tag{9}$$

where the second equality is an eigendecomposition of $\mathbf{J}$, $\mathbf{\Psi}$ is the matrix of eigenvectors, and $\mathbf{D}_\mu$ is the diagonal matrix of eigenvalues μ of $\mathbf{J}$. The second step computes a set of sparse coefficients α that select the active modes in the data. This is achieved by solving the following sparse coding problem

$$\min_\alpha \|\mathbf{Y}_1 - \mathbf{U}\mathbf{\Psi}\mathbf{D}_\alpha\mathbf{V}_\mu\|_F + \gamma\|\alpha\|_1, \tag{10}$$

where $\mathbf{D}_\alpha$ is a diagonal matrix with α on the diagonal, and $\mathbf{V}_\mu$ is the Vandermond matrix of the vector μ. The sp-DMD approach works quite well where the data is contaminated with relatively small noise. However, its performance suffers when large noise, or outliers are present in the data. This is primarily due to the PCA dimensionality reduction applied in the first step of the algorithm resulting in noisy modes. Therefore, the second step cannot find a sparse-enough subset to explain the observations.

Total least squares DMD (tls-DMD) [8] is another recent method that strives to eliminate noise and find the true poles (eigenvalues) of the DMD operator from the data. In a nutshell it obtains better accuracy by applying a simple lifting to the data and partial SVD. Consider the lifted data matrix $\mathbf{Y}_{\text{lift}} = [\mathbf{Y}_1^T, \mathbf{Y}_2^T]^T$ in $2N$ dimensional space, the tls-DMD method first computes an SVD of $\mathbf{Y}_{\text{lift}} = \bar{\mathbf{U}}\bar{\Sigma}\bar{\mathbf{V}}^T$. The matrix $\bar{\mathbf{U}}$ is then restricted to the first r columns and split into two submatrices $\bar{\mathbf{U}}_r = [\bar{\mathbf{U}}_{1,r}^T, \bar{\mathbf{U}}_{2,r}^T]^T$, where $\bar{\mathbf{U}}_{1,r}$ and $\bar{\mathbf{U}}_{2,r}$ is each of size $N \times r$. The tls-DMD operator is then computed as

$$\mathbf{K}_{tls} = \bar{\mathbf{U}}_{r,2}\bar{\mathbf{U}}_{r,1}^{\dagger}. \tag{11}$$

This method is effective in the cases where the order of the underlying system is known and when the noise is white Gaussian distributed. However, it also suffers when the data is contaminated with outliers.

5. FACTORIZED LOW-RANK DMD

We now introduce our robust DMD algorithm. Suppose that there exist noise-free velocity observations $\mathbf{y}_k$ measuring the true motion of pedestrians in the scene and let the motion vectors $\mathbf{z}_k$ correspond to noisy observations such that

$$\mathbf{z}_k + \mathbf{e}_k = \mathbf{y}_k, \tag{12}$$

where $\mathbf{e}_k$ is the additive noise. Let $\mathbf{E}_1, \mathbf{E}2$, $\mathbf{Z}_1$, $\mathbf{Z}_2$ and $\mathbf{Y}_1, \mathbf{Y}_2$ be as in (5) for $\mathbf{e}_k$, $\mathbf{z}_k$ and $\mathbf{y}_k$, respectively. Then the Koopman operator corresponding to the noise-free dynamics is given by the total least squares constraint

$$\begin{aligned} \mathbf{Y}_2 &= \mathbf{K}\mathbf{Y}_1 \\ \Leftrightarrow \quad \mathbf{Z}_2 + \mathbf{E}2 &= \mathbf{K}\left(\mathbf{Z}_1 + \mathbf{E}_1\right). \end{aligned} \tag{13}$$

However, problem (13) is non-convex and ill-posed since only $\mathbf{Z}_1$ and $\mathbf{Z}_2$ are observed and $\mathbf{K}$, $\mathbf{E}_1$ and $\mathbf{E}_2$ are unknowns. To remedy this situation, we invoke the following priors in our problem. Highly complex systems can be accurately modeled by low-order dynamics. This translates into the prior that the operator $\mathbf{K}$ modeling the evolution of a noise-free system has a low rank. The second prior derives from the definition of the matrices $\mathbf{E}_1$ and $\mathbf{E}_2$. Denote by $\mathbf{I}_{m-1}$ the identity matrix of size $(m-1) \times (m-1)$, and let $\mathbb{I}^0 = [\mathbf{0}|\mathbf{I}_{m-1}]^T$ and $\mathbb{I}_0 = [\mathbf{I}_{m-1}|\mathbf{0}]^T$ be selection operators that respectively remove the first column and the last column of a matrix with m columns. Then, $\mathbf{E}_1$ and $\mathbf{E}_2$ satisfy the relation

$$\mathbf{E}_1\mathbb{I}^0 = \mathbf{E}_2\mathbb{I}_0. \tag{14}$$

We pose the low-rank dynamic mode decomposition problem with total least squares constraints as follows

$$\begin{aligned} \min_{\mathbf{K},\mathbf{E}_1,\mathbf{E}_2} \quad & \|\mathbf{K}\|_* + \tfrac{\gamma}{2}\left(\|\mathbf{E}_2\|_F^2 + \|\mathbf{e}_0\|_2^2\right) \\ \text{subject to} \quad & \mathbf{Z}_2 + \mathbf{E}2 = \mathbf{K}\left(\mathbf{Z}_1 + \mathbf{E}_1\right) \\ & \mathbf{E}_1\mathbb{I}^0 = \mathbf{E}_2\mathbb{I}_0, \end{aligned} \tag{15}$$

where $\mathbf{e}_0$ is the first column in $\mathbf{E}_1$, γ is a regularization parameter, and $\|\mathbf{K}\|_*$ is the nuclear norm equal to the sum of the singular values of a matrix $\mathbf{K}$.

Let N be the number of motion blocks in a video frame. Then the size of the operator $\mathbf{K}$ will be equal to $N \times N$ which can become prohibitively expensive to store and compute for high resolution videos. Therefore, we replace $\mathbf{K}$ with its rank r factors $\mathbf{L} \in \mathbb{R}^{N\times r}$ and $\mathbf{R} \in \mathbb{R}^{N\times r}$, such that $\mathbf{K} = \mathbf{L}\mathbf{R}^T$ to reduce the computational complexity. Moreover, we employ the nuclear norm proxy proposed in [20] which replaces the nuclear norm of a matrix with the average of the square of the Frobenius norms of its low rank factors

$$\|\mathbf{K}\|_* = \inf_{\mathbf{L},\mathbf{R}:\mathbf{K}=\mathbf{L}\mathbf{R}^T} \frac{1}{2}\left(\|\mathbf{L}\|_F^2 + \|\mathbf{R}\|_F^2\right). \tag{16}$$

Consequently, the factorized low-rank DMD problem with total least squares constraints (flora-DMD-tls) is formulated as follows

$$\begin{aligned} \min_{\mathbf{L},\mathbf{R},\mathbf{E}_1,\mathbf{E}_2} \quad & \tfrac{1}{2}\left(\|\mathbf{L}\|_F^2 + \|\mathbf{R}\|_F^2\right) + \tfrac{\gamma}{2}\left(\|\mathbf{E}_2\|_F^2 + \|\mathbf{e}_0\|_2^2\right) \\ \text{subject to} \quad & \mathbf{Z}_2 + \mathbf{E}_2 = \mathbf{L}\mathbf{R}^T\left(\mathbf{Z}_1 + \mathbf{E}_1\right) \\ & \mathbf{E}_1\mathbb{I}^0 = \mathbf{E}_2\mathbb{I}_0, \end{aligned} \tag{17}$$

which we solve using an alternating direction method of multiplier (ADMM) algorithm formulated below in unconstrained form.

$$\begin{aligned} \max_{\mathbf{P},\mathbf{Q},\mathbf{W}} \min_{\mathbf{L},\mathbf{R},\mathbf{C},\mathbf{E}_1,\mathbf{E}_2} \quad & \tfrac{1}{2}\|\mathbf{L}\|_F^2 + \tfrac{1}{2}\|\mathbf{R}\|_F^2 + \tfrac{\gamma}{2}\left(\|\mathbf{E}_2\|_F^2 + \|\mathbf{e}_0\|_2^2\right) \\ & + \tfrac{\mu}{2}\|\mathbf{Z}_2 + \mathbf{E}_2 - \mathbf{L}\mathbf{C} + \mathbf{P}/\mu\|_F^2 \\ & + \tfrac{\eta}{2}\|\mathbf{C} - \mathbf{R}^T\left(\mathbf{Z}_1 + \mathbf{E}_1\right) + \mathbf{W}/\eta\|_F^2 \\ & + \tfrac{\rho}{2}\|\mathbf{E}_1\mathbb{I}^0 - \mathbf{E}_2\mathbb{I}_0 + \mathbf{Q}/\rho\|_F^2 \end{aligned} \tag{18}$$

The update steps of (18) are shown in detail in Algorithm 1. Here we introduced the temporary variable $\mathbf{C}$ to simplify the update steps of $\mathbf{L}$ and $\mathbf{R}$. The variables $\mathbf{P}$, $\mathbf{Q}$, $\mathbf{W}$ are Lagrange multipliers, and μ, η, and ρ are the augmented Lagrangian parameters.

Algorithm 1 Factorized low-rank DMD with total least squares (flora-DMD-tls)

1: **Input** $\mathbf{Z}_1$, $\mathbf{Z}_2$, γ, $\mathbf{R}$, `maxiter`, $c > 1$
2: **Output** $\mathbf{L}$, $\mathbf{R}$, $\mathbf{E}_1$, $\mathbf{E}_2$
3: **Initialize** $j = 0$, $\mu = 1$, $\eta = 1$, $\rho = 1$, $\mathbf{C} = \mathbf{R}^T(\mathbf{Z}_1)$, $\mathbf{E}_1 = \mathbf{0}$, $\mathbf{E}_2 = \mathbf{0}$, $\mathbf{P} = \mathbf{0}$, $\mathbf{Q} = \mathbf{0}$, $\mathbf{W} = \mathbf{0}$
4: **for** $j = 0$ **to** `maxiter` **do**
5: $\quad \mathbf{L} = (\mathbf{P} + \mu(\mathbf{Z}_2 + \mathbf{E}_2))\mathbf{C}^T\left(\mathbf{I}_r + \mu\mathbf{C}\mathbf{C}^T\right)^{-1}$
6: $\quad \mathbf{C} = (\eta\mathbf{I}_r + \mu\mathbf{L}^T\mathbf{L})^{-1}\left(\mathbf{L}^T(\mathbf{P} + \mu(\mathbf{Z}_2 + \mathbf{E}_2)) + \eta\mathbf{R}^T(\mathbf{Z}_1 + \mathbf{E}_1) - \mathbf{W}\right)$
7: $\quad \mathbf{R} = \left(\mathbf{I}_N + \eta(\mathbf{Z}_1 + \mathbf{E}_1)(\mathbf{Z}_1 + \mathbf{E}_1)^T\right)^{-1}(\mathbf{Z}_1 + \mathbf{E}_1)(\eta\mathbf{C} + \mathbf{W})^T$
8: $\quad \mathbf{E}_2\mathbb{I}_0 = \left(-\mu(\mathbf{Z}_2 - \mathbf{L}\mathbf{C} + \frac{1}{\mu}\mathbf{P}) + (\rho\mathbf{E}_1\mathbb{I}^0 - \mathbf{Q})\mathbb{I}_0^T\right)/(\gamma + \mu + \rho)$
9: $\quad \mathbf{e}_m = -\mu(\mathbf{Z}_2 - \mathbf{L}\mathbf{C} + \frac{1}{\mu}\mathbf{P})/(\gamma + \mu)$
10: $\quad \mathbf{E}_1\mathbb{I}^0 = (\rho\mathbf{I}_N + \eta\mathbf{R}\mathbf{R}^T)^{-1}\left(\eta\mathbf{R}(\mathbf{C} - \mathbf{R}^T\mathbf{Z}_1 + \frac{1}{\eta}\mathbf{W}) + \rho(\mathbf{E}_2\mathbb{I}_0 + \mathbf{Q}/\rho)\mathbb{I}^{0T}\right)$
11: $\quad \mathbf{e}_0 = (\gamma\mathbf{I}_N + \eta\mathbf{R}\mathbf{R}^T)^{-1}\left(\eta\mathbf{R}(\mathbf{C} - \mathbf{R}^T\mathbf{Z}_1 + \frac{1}{\eta}\mathbf{W})\right)$
12: $\quad \mathbf{P} = \mathbf{P} + \mu(\mathbf{Z}_2 + \mathbf{E}_2 - \mathbf{L}\mathbf{C})$
13: $\quad \mathbf{W} = \mathbf{W} + \eta(\mathbf{C} - \mathbf{R}^T(\mathbf{Z}_1 + \mathbf{E}_1))$
14: $\quad \mathbf{Q} = \mathbf{Q} + \rho(\mathbf{E}_1\mathbb{I}^0 - \mathbf{E}_2\mathbb{I}_0)$
15: $\quad \mu = c\mu, \eta = c\eta, \rho = c\rho$
16: **end for**

Steps 7, 10, and 11 in Algorithm 1 involve inverting large $N \times N$ matrices, which can be prohibitive when $N \gg m$. Instead, we utilize the matrix inversion lemma to reduce the complexity of the

computation. The corresponding update steps are shown below:

$$\begin{aligned}&\mathbf{D}_R = (\tfrac{1}{\eta}\mathbf{I}_m + (\mathbf{Z}_1+\mathbf{E}_1)^T(\mathbf{Z}_1+\mathbf{E}_1)),\\ &\mathbf{R} = \big(\mathbf{I}_N - (\mathbf{Z}_1+\mathbf{E}_1)\mathbf{D}_R^{-1}(\mathbf{Z}_1+\mathbf{E}_1)^T\big)(\mathbf{Z}_1+\mathbf{E}_1)(\eta\mathbf{C}+\mathbf{W})^T\end{aligned} \tag{19}$$

$$\begin{aligned}\mathbf{D}_E &= \rho^2(\tfrac{1}{\eta}\mathbf{I}_r + \tfrac{1}{\rho}\mathbf{R}^T\mathbf{R}),\\ \mathbf{E}_1\mathbb{I}^0 &= \big(\tfrac{1}{\rho}\mathbf{I}_N - \mathbf{R}\mathbf{D}_E^{-1}\mathbf{R}^T\big)\big(\eta\mathbf{R}(\mathbf{C}-\mathbf{R}^T\mathbf{Z}_1+\tfrac{1}{\eta}\mathbf{W})\\ &\quad +\rho(\mathbf{E}_2\mathbb{I}_0+\mathbf{Q}/\rho)\mathbb{I}^{0T}\big)\end{aligned} \tag{20}$$

$$\begin{aligned}&\mathbf{D}_e = \gamma^2(\tfrac{1}{\eta}\mathbf{I}_r + \tfrac{1}{\gamma}\mathbf{R}^T\mathbf{R}),\\ &\mathbf{E}_1\mathbb{I}^0 = \big(\tfrac{1}{\gamma}\mathbf{I}_N - \mathbf{R}\mathbf{D}_e^{-1}\mathbf{R}^T\big)\big(\eta\mathbf{R}(\mathbf{C}-\mathbf{R}^T\mathbf{Z}_1+\tfrac{1}{\eta}\mathbf{W})\big)\end{aligned} \tag{21}$$

The problem defined in (17) addresses the case where the noise is Gaussian distributed. If in addition, the measurements $\mathbf{Z}$ are contaminated with outlier sparse noise, we propose the following robust formulation to the problem

$$\begin{aligned}\min_{\mathbf{L},\mathbf{R},\mathbf{E}_1,\mathbf{E}_2,\mathbf{S}_1,\mathbf{S}_2}\quad & \tfrac{1}{2}\left(\|\mathbf{L}\|_F^2+\|\mathbf{R}\|_F^2\right)\\ &+\tfrac{\gamma}{2}\left(\|\mathbf{E}_2\|_F^2+\|\mathbf{e}_0\|_2^2\right)+\lambda\|\mathbf{S}_2\|_1\\ \text{subject to}\quad & \mathbf{Z}_2+\mathbf{E}_2+\mathbf{S}_2 = \mathbf{L}\mathbf{R}^T(\mathbf{Z}_1+\mathbf{E}_1+\mathbf{S}_1)\\ & \mathbf{E}_1\mathbb{I}^0 = \mathbf{E}_2\mathbb{I}_0\\ & \mathbf{S}_1\mathbb{I}^0 = \mathbf{S}_2\mathbb{I}_0,\end{aligned} \tag{22}$$

where the variables $\mathbf{S}_1$ and $\mathbf{S}_2$ capture the outlier noise in the data, and γ and λ are regularization parameters. The solution to (22) can also be obtained using an ADMM algorithm similar to Algorithm 1 which we omit from the presentation. We refer to problem (22) as *robust flora-DMD-tls*.

The total least squares constraints in both *flora-DMD-tls* (17) and *robust flora-DMD-tls* (22) represent exact, albeit non-convex constrained, modeling of the observed data. We evaluate the validity of these models in the next section by comparing with the convex constrained problem

$$\begin{aligned}\min_{\mathbf{L},\mathbf{R},\mathbf{E}_2}\quad & \tfrac{1}{2}\left(\|\mathbf{L}\|_F^2+\|\mathbf{R}\|_F^2\right)+\tfrac{\gamma}{2}\|\mathbf{E}_2\|_F^2\\ \text{subject to}\quad & \mathbf{Z}_2+\mathbf{E}_2 = \mathbf{L}\mathbf{R}^T\mathbf{Z}_1,\end{aligned} \tag{23}$$

where the $\mathbf{E}_1$ term is dropped from the optimization. We refer to problem (23) as *flora-DMD-ls*.

6. NUMERICAL RESULTS

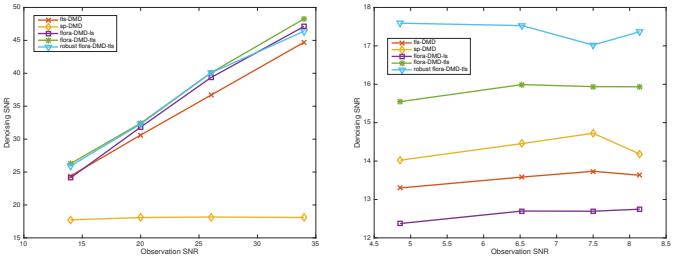
We test the performance of our proposed algorithms on synthetic data and on real crowd and traffic videos. As a benchmark, we compare the denoising performance and the prediction performance with that of *sp-DMD* and *tls-DMD* using the synthetic dataset. Then we apply our *robust flora-DMD-tls* algorithm to compressed domain motion vectors extracted from HEVC [21] encoded video sequences from the UCF crowd analysis dataset [10].

6.1. Synthetic data

We generate a multidimensional time series $\mathbf{Y}$ of length 100 with kth column $\mathbf{y}_k \in \mathbb{R}^{200}$ driven by a rank 10 one-step prediction operator $\mathbf{K}$ with eigenvalues bounded in magnitude by the interval $[0.3, 1.01]$. The vector $\mathbf{y}_1$ is initialized with independent identically distributed (i.i.d.) standard Gaussian random entries drawn from $\mathcal{N}(0,1)$. The series $\{\mathbf{y}_k\}$ is then computed using the recursion $\mathbf{y}_k = \mathbf{K}\mathbf{y}_{k-1}$ for $k \in \{2\dots 100\}$.

Next, we generate noisy observations $\mathbf{z}_k = \mathbf{y}_k + \mathbf{e}_k + \mathbf{s}_k$ by adding dense white Gaussian noise $\mathbf{e}_k$ with a signal-to-noise-ratio (SNR) of 14dB, 20dB, 26dB, and 34dB, as well as sparse outliers $\mathbf{s}_k$ to the data. The outliers are set such that 10% of the data entries are corrupted with i.i.d. entries drawn from $\mathcal{N}(0,4)$. For each noise setting, we generate 20 instances and run standard *DMD*, *tls-DMD*, *sp-DMD*, *flora-DMD-ls*, *flora-DMD-tls*, and *robust flora-DMD-tls* to denoise the data and/or recovery the operator K. Except for *sp-DMD*, we perform the recovery while overestimating the rank by setting the rank of the factors to 16. We have included the recovery results for when the factors have a correct rank of 10 in the supplementary material. In the *flora-DMD* algorithms, we set the parameters $\lambda = 0.1$ and choose the best performance between $\gamma = 1, 5$ and 10. In *sp-DMD*, we set $\gamma = 200$ which we found to perform best in terms of recovery performance. Fig. 1 and Fig. 2 compare the recovery performance of all algorithms in terms of the denoising SNR and prediction SNR, respectively, averaged over the 20 instantiations of the noise. Here we define the error in denoising for the *flora-DMD* algorithms as the difference $\mathbf{y}_k - (\mathbf{z}_k + \hat{\mathbf{e}}_k + \hat{\mathbf{s}}_k)$, where $\hat{\mathbf{e}}_k$ and $\hat{\mathbf{s}}_k$ are the recovered noise and outliers. For *sp-DMD*, the recovered error is the difference between $\mathbf{Y}$ and the recovered signal $\widehat{\mathbf{Y}}_{\text{sp}} = \mathbf{U}\mathbf{\Psi}\mathbf{D}_{\alpha^*}\mathbf{V}_\mu$, where α^* is the solution of (10). On the other hand, we use the rank-r truncated lifted data matrix to form the *tls-DMD* denoised data matrix $\widehat{\mathbf{Y}}_{\text{tls}} = \bar{\mathbf{U}}_{r,2}\bar{\Sigma}\bar{\mathbf{V}}^T$.

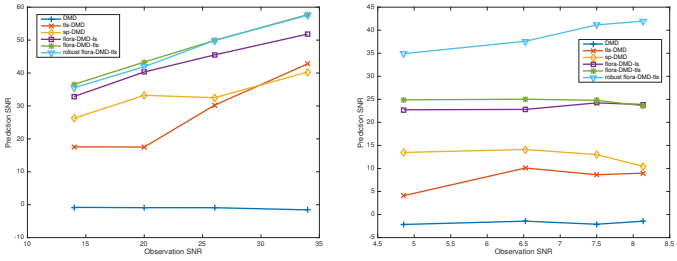
The prediction error is computed by driving the noise free system using the operator $\mathbf{K}$ for another 100 time steps to produce a ground truth prediction matrix $\mathbf{Y}_p$ of size 200×100. For each of the six algorithms defined above, we use the respective estimated DMD operator to drive the system for 100 time steps using the same initial vector $\mathbf{y}_{100}$. All six algorithms allow prediction and the error in the prediction is then computed relative to $\mathbf{Y}_p$. The experiments show that the *flora-DMD* algorithms result in better denoising and prediction performance compared to *sp-DMD* and *tls-DMD*. Moreover, in the absence of outliers, *flora-DMD-tls* seems to perform best. On the other hand, the existence of outliers dominates the performance of the algorithms favoring *robust flora-DMD-tls* over the others.



(a) (b)

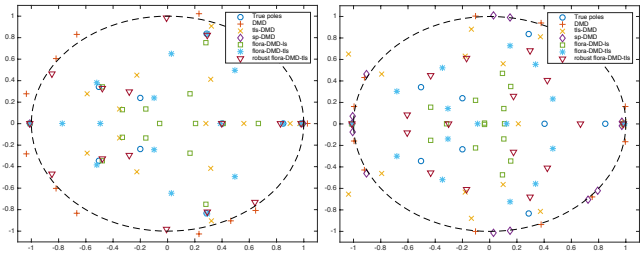
Fig. 1: Comparison of denoising performance when the data have (a) no outliers, and (b) 10% of the entries are corrupted by outliers.

Next we show in Fig. 3 the performance of the six algorithms at recovering the true poles of the operator $\mathbf{K}$ from 20dB SNR data and with or without outliers. Notice that in all cases, the large magnitude poles are captured by all algorithms. When the data contains no outliers, *flora-DMD-tls* does a better job at recovering the correct locations of the poles, including the transient (small magnitude) poles. On the other hand, the introduction of outliers makes it harder to capture the correct pole locations. However, the improved denoising and prediction performance of *robust flora-DMD-tls* compared to the other algorithms leads us to believe that true poles and modes lie in the span of the estimated poles and modes.



(a) (b)

Fig. 2: Comparison of prediction performance when the data have (a) no outliers, and (b) 10% of the entries are corrupted by outliers.



(a) (b)

Fig. 3: Estimated locations of the poles of the DMD operator when the data are corrupted by 20dB SNR and have (a) no outliers, and (b) 10% of the entries are corrupted by outliers.

6.2. Crowd/traffic videos

Next, we use our *robust flora-DMD-tls* algorithm to identify steady-state and transient flows in real video sequences. For this purpose, we select two sequences from the UCF crowd analysis dataset, one of a crowd flow and the other of a chaotic traffic flow. We first form the data matrices $\mathbf{Z}_1$ and $\mathbf{Z}_2$ by vertically stacking the horizontal and vertical motion vectors of each frame indexed by k into a vector $\mathbf{z}_k$ as discussed in Section 4.2. Therefore, a video sequence containing N motion blocks and $T+1$ video frames produces a data matrix of size $2N \times T$. Assuming that the motion vectors are noisy observations of the velocity field of an underlying dynamical system, we estimate the Koopman operator $\mathbf{K}_{\text{flora}} = \mathbf{L}\mathbf{R}^T$ and perform an eigendecomposition of $\mathbf{K}_{\text{flora}=\mathbf{U}\mathbf{\Lambda}\mathbf{U}^\dagger}$, where $\mathbf{U}$ is the eigenvector matrix and $\mathbf{\Lambda}$ is the diagonal matrix containing the eigenvalues of $\mathbf{K}$. For all videos we set an operator rank equal to 10. Hence, $\mathbf{K}_{\text{flora}}$ has 10 eigenvectors and 10 eigenvalues.

Our flow analysis is based on separating the flow directions specified by the columns of $\mathbf{U}$ according to the magnitudes of the associated eigenvalues in Λ. Therefore, we split the range of magnitudes between $(0, 1.2]$ into six segments of width 0.2. Next, we compute the mean flow $\overline{\mathbf{Z}}$ by averaging across the rows of $\mathbf{Z}_2$ and evaluate the projection coefficients $\overline{\alpha} = \mathbf{U}^\dagger\overline{\mathbf{Z}}$ of $\overline{\mathbf{Z}}$ onto $\mathbf{U}$. For each segment indexed by $j \in \{1, \dots 5\}$, we can now compute an eigenflow $\mathbf{f}_j \in \mathbb{R}^{2N}$ as follows

$$\mathbf{f}_j = \mathbf{U}_{\Omega_j}\mathbf{\Lambda}_{\Omega_j}\overline{\alpha}_{\Omega_j}, \tag{24}$$

where Ω_j is the set of indices corresponding to the eigenvalues and associated eigenvectors having with magnitudes in segment j. The first N entries in $\mathbf{f}_j$ contain the horizontal eigenmotionvectors, while the remaining N entries contain the vertical eigen-motionvectors.

Figs. 4 and 5 show the transient and steady-state flows identified by our method when applied to video sequences from the UCF dataset, namely, Seq31 and Seq20. The figures also show the poles of the estimated Koopman operators for each video. The first two transient flows of Seq 31 illustrated in Fig.4 (b) and (c) correspond to the camera shake. The third transient flow in Fig.4 (d) captures the motion of people crossing between the dominant steady state flows captured in Fig.4 (e). Similarly in Seq20, the transient flows of Fig.5 (b) and (c) are also dominated by the camera shake, while the flow in (c) also captures some transients in the traffic. The flow in Fig.5 (d) is mainly dominated by the motion of vehicles moving to the bottom left of the screen, whereas the stead-state flow captures the turning vehicles as well as the the straight motion towards the top right of the screen. The complete videos overlaid with the input motion vectors are available in the supplementary material.

7. CONCLUSION

We developed a factorized low rank dynamic mode decomposition algorithm (*flora-DMD*) that is robust to both inlier and outlier noise in data. Our method estimates the Koopman modes of the data dynamics by formulating a rank regularized DMD problem with total least squares constraints and solving the problem using an ADMM algorithm. We demonstrated using experiments on synthetic data that our method enjoys better data denoising and prediction capabilities compared to the recently proposed *sp-DMD* and *tls-DMD*. We also demonstrate the effectiveness of our approach in crowd and traffic flow analysis. Using only noisy compressed domain motion vectors as input, our method is capable of denoising the motion vectors and determining steady-state and transient flows in the video sequences.

8. REFERENCES

[1] K. Cao, Y. Chen, D. Stuart, and D. Yue, "Cyber-physical modeling and control of crowd of pedestrians: A review and new framework," *IEEE/CAA Journal of Automatica Sinica*, vol. 2, no. 3, pp. 334, 2015.

[2] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson, "Spectral analysis of nonlinear flows," *Journal of Fluid Mechanics*, vol. 641, pp. 115–127, 12 2009.

[3] PJ Schmid, "Dynamic mode decomposition of numerical and experimental data," *Journal of Fluid Mechanics*, vol. 656, pp. 5–28, 2010.

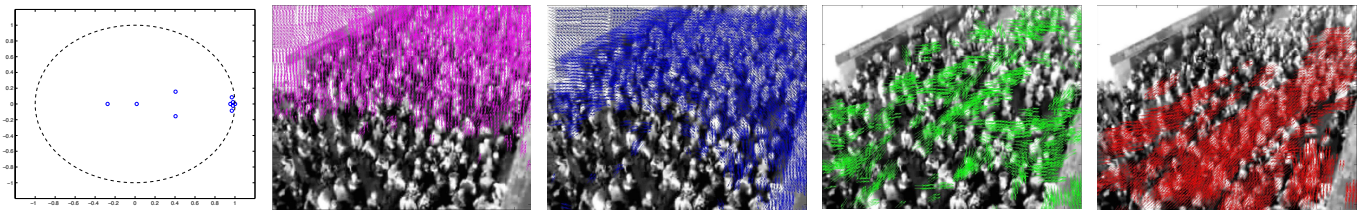
[4] I. Mezić and A. Banaszuk, "Comparison of systems with complex behavior," *Physica D Nonlinear Phenomena*, vol. 197, pp. 101–133, Oct. 2004.

[5] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, "On dynamic mode decomposition: Theory and applications," *Journal of Computational Dynamics*, vol. 1, no. 2, pp. 391–421, 2014.

[6] J. Grosek and J. N. Kutz, "Dynamic mode decomposition for real-time background/foreground separation in video," *CoRR*, vol. abs/1404.7592, 2014.

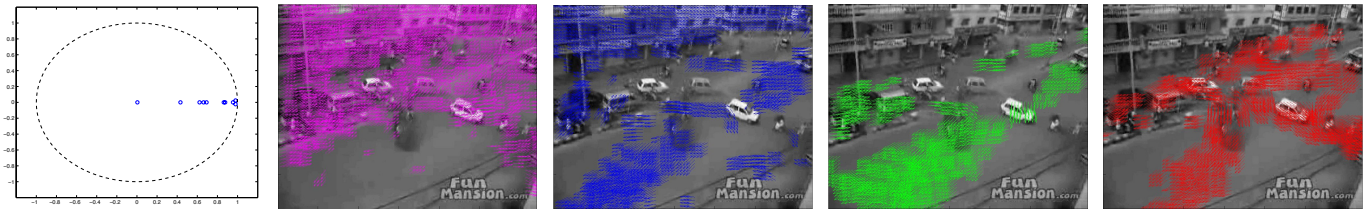
[7] M. R. Jovanović, P. J Schmid, and J. W. Nichols, "Sparsity-promoting dynamic mode decomposition," *Physics of Fluids (1994-present)*, vol. 26, no. 2, pp. 024103, 2014.

[8] S. Dawson, M. Hemati, M. Williams, and C. Rowley, "Characterizing and correcting for the effect of sensor noise in the dynamic mode decomposition," *Bulletin of the American Physical Society*, vol. 59, 2014.



(a) (b) (c) (d) (e)

Fig. 4: Detected eigenflows for sequence number 31 from the UCF crowd dataset. (a) Positions of the computed eigenvalues, (b) transient flow of segment $(0, 0.2]$, (c) transient flow of segment $(0.4, 0.6]$, (d) transient flow of segment $(0.8, 1)$, (e) stead-state flow.



(a) (b) (c) (d) (e)

Fig. 5: Detected eigenflows for sequence number 20 from the UCF crowd dataset. (a) Positions of the computed eigenvalues, (b) transient flow of segment $(0, 0.2]$, (c) transient flow of segment $(0.4, 0.6]$, (d) transient flow of segment $(0.6, 0.8)$, (e) stead-state flow.

[9] S. Ali and M. Shah, "A Lagrangian particle dynamics approach for crowd flow segmentation and stability analysis," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–6.

[10] B. Solmaz, B. E. Moore, and M. Shah, "Identifying behaviors in crowd scenes using stability analysis for dynamical systems," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 10, pp. 2064–2070, 2012.

[11] B. T. Morris and M. M. Trivedi, "Trajectory learning for activity understanding: Unsupervised, multilevel, and long-term adaptive approach," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 11, pp. 2287–2301, 2011.

[12] K. Kim, D. Lee, and I. Essa, "Gaussian process regression flow for analysis of motion trajectories," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1164–1171.

[13] A. Nakhmani, A. Surana, and A. Tannenbaum, "Macroscopic analysis of crowd motion in video sequences," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. IEEE, 2014, pp. 1822–1827.

[14] F. Zhu, X. Wang, and N. Yu, "Crowd tracking with dynamic evolution of group structures," in *Computer Vision–ECCV 2014*, pp. 139–154. Springer, 2014.

[15] I. Saleemi, L. Hartung, and M. Shah, "Scene understanding by statistical modeling of motion patterns," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2069–2076.

[16] D. Lin, E. Grimson, and J. Fisher, "Learning visual flows: A Lie algebraic approach," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 747–754.

[17] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto, "Dynamic textures," *International Journal of Computer Vision*, vol. 51, no. 2, pp. 91–109, 2003.

[18] A. B. Chan and N. Vasconcelos, "Modeling, clustering, and segmenting video with mixtures of dynamic textures," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 5, pp. 909–926, 2008.

[19] R. L. Hughes, "A continuum theory for the flow of pedestrians," *Transportation Research Part B: Methodological*, vol. 36, no. 6, pp. 507 – 535, 2002.

[20] N. Srebro, *Learning with matrix factorizations*, Ph.D. thesis, Cambridge, MA, USA, 2004, AAI0807530.

[21] B. Bross, W. J. Han, J. R. Ohm, G. J. Sullivan, Y. K. Wang, and T. Wiegand, *High Efficiency Video Coding (HEVC) text specification draft 10*, JCT-VC of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Jan. 2013.