MULTI-RESOLUTION TIME-SCALE SEPARATION OF VIDEO CONTENT USING THE DYNAMIC MODE DECOMPOSITION

J. Nathan Kutz, J. Grosek[*]*, X. Fu and S. Brunton*[†]

Department of Applied Mathematics, University of Washington, Seattle, WA 98195-3925
[*]Air Force Research Laboratories, Albuquerque, NM
[†]Department of Mechanical Engineering, University of Washington, Seattle, WA 98195

ABSTRACT

We introduce the method of dynamic mode decomposition (DMD) for robustly separating video frames into a hierarchy of multi-resolution time-scaled components. The method includes a methodology for background (low-rank) and foreground (sparse) separation of video content in real-time. The method involves the application of a technique used for characterizing nonlinear dynamical systems in an equation-free manner by decomposing the state of the system into low-rank terms whose Fourier components in time are known. Thus the method integrates two of the leading data analysis methods in use today: Fourier transforms and Principal Components. DMD terms with Fourier frequencies near the origin (zero-modes) are interpreted as background (low-rank) portions of the given video frames, and the terms with Fourier frequencies bounded away from the origin are their sparse counterparts. An approximate low-rank/sparse separation is achieved at the computational cost of just one singular value decomposition and one linear equation solve, thus producing results orders of magnitude faster than a leading separation method, namely robust principal component analysis (RPCA). The DMD method that is developed here is demonstrated to work robustly in real-time with personal laptop-class computing power and without any parameter tuning, which is a transformative improvement in performance that is ideal for video surveillance and recognition applications. By applying it recursively in a multi-resolution manner, a time-frequency analysis can be made of multiple time-scales in video.

1. INTRODUCTION

Accurate and real-time video surveillance techniques for removing *background* variations in a video stream, which are highly correlated between frames, are at the forefront of modern data-analysis research. The objective in such algorithms is to highlight *foreground* objects of potential interest . Background/foreground separation is typically an integral step in detecting, identifying, tracking, and recognizing objects in video sequences. Most modern computer vision applications demand algorithms that can be implemented in real-time, and that are robust enough to handle diverse, complicated, and cluttered backgrounds. Competitive methods often need to be flexible enough to accommodate changes in a scene due to, for instance, illumination changes that can occur throughout the day, or location changes where the application is being implemented. Given the importance of this task, a variety of iterative techniques and methods have already been developed in order to perform background/foreground separation [1, 2, 3, 4, 5] (See also, for instance, the recent review [6] which compares error and timing of various methods).

One potential viewpoint of this computational task is as a matrix separation problem into *low-rank* (background) and *sparse* (foreground) components. Recently, this viewpoint has been advocated by Candès et al. in the framework of *robust principal component analysis* (RPCA) [5]. By weighting a combination of the nuclear and the L^1 norms, a convenient convex optimization problem (*principal component pursuit*) was demonstrated, under suitable assumptions, to recover the low-rank and sparse components exactly of a given data-matrix (or video for our purposes). It was also compared to the state-of-the-art computer vision procedure developed by De La Torre and Black [7]. We advocate a similar matrix separation approach, but by using the method of *dynamic mode decomposition* (DMD) [8, 9, 10, 11, 12, 13]. This method, which essentially implements a Fourier decomposition of the video frames in time, distinguishes the stationary background from the dynamic foreground by differentiating between the near-zero modes and the remaining modes bounded away from the origin, respectively. Originally introduced in the fluid mechanics community, DMD has emerged as a powerful tool for analyzing the dynamics of nonlinear systems [8, 9, 10, 11, 12, 13].

In the application of video surveillance, the video frames can be thought of as snapshots of some underlying complex/nonlinear dynamics. The DMD decomposition yields

J.N.K. Acknowledges support from the U.S. Air Force Office of Scientific Research (FA9550-09-0174).

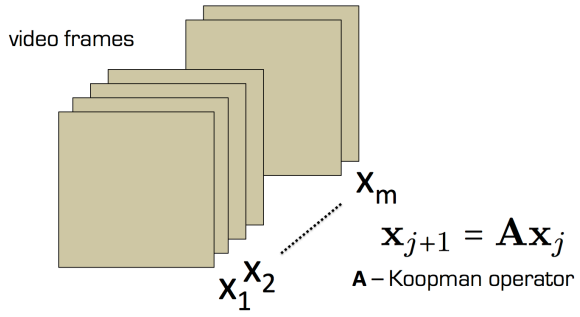

Fig. 1. Illustration of the DMD method where snapshots (video frames) $\mathbf{x}_j$ are taken and a linear transformation $\mathbf{A}$ (Koopman operator) is constructed. The DMD method constructs the best matrix $\mathbf{A}$ that minimizes the least-square error for all transformations $\mathbf{x}_{j+1} = \mathbf{A}\mathbf{x}_j$ with $j = 1, 2, \cdots, m-1$.

oscillatory time components of the video frames that have contextual implications. Namely, those modes that are near the origin represent dynamics that are unchanging, or changing slowly, and can be interpreted as stationary background pixels, or low-rank components of the data matrix. In contrast, those modes bounded away from the origin are changing on $\mathcal{O}(1)$ timescales or faster, and represent the foreground motion in the video, or the sparse components of the data matrix. Thus, by simply applying the dynamical systems DMD interpretation to video frames, an approximate RPCA technique can be enacted at a fixed cost of a singular-value decomposition and a linear equation solve. The innovation of multi-resolution DMD (MRDMD) allows for further separation of dynamic content in the video, thus allowing for the separation of components that are happening on different time scales.

2. DYNAMIC MODE DECOMPOSITION

The DMD method provides a spatio-temporal decomposition of data into a set of dynamic modes that are derived from snap shots or measurements of a given system in time. The mathematics underlying the extraction of dynamic information from time-resolved snapshots is closely related to the idea of the Arnoldi algorithm, one of the workhorses of fast computational solvers. The data collection process involves two parameters:

$$N = \text{number of pixels saved per snapshot} \quad (1a)$$
$$M = \text{number of snapshots taken} \quad (1b)$$

Originally the algorithm was designed to collect data at regularly spaced intervals of time. However, new innovations allow for both sparse spatial and temporal collection of data as well as irregularly spaced collection times. To illustrate the algorithm, we consider regularly spaced sampling in time:

$$\text{data collection times}: \quad t_{m+1} = t_m + \Delta t \qquad (2)$$

where the collection time starts at t_1 and ends at t_M, and the interval between data collection times is Δt. In the MRDMD method, the number of snapshots will vary as the algorithm extracts multi-timescale spatio-temporal structures. This will be the central focus of the next section.

The data snapshots are arranged into an $N \times M$ matrix

$$\mathbf{X} = [\mathbf{x}(t_1)\ \mathbf{x}(t_2)\ \mathbf{x}(t_3)\ \cdots\ \mathbf{x}(t_M)] \qquad (3)$$

where the vector $\mathbf{x}$ are the N measurements of the state variable of the system of interest at the data collection points. The objective is to mine the data matrix $\mathbf{X}$ for important dynamical information. For the purposes of the DMD method, the following matrix is also defined:

$$\mathbf{X}_j^k = [\mathbf{x}(t_j)\ \mathbf{x}(t_{j+1})\ \cdots\ \mathbf{x}(t_k)] \qquad (4)$$

Thus this matrix includes columns j through k of the original data matrix.

The DMD method approximates the modes of the so-called *Koopman operator*. The Koopman operator is a linear, infinite-dimentional operator that represents nonlinear dynamics without linearization [11], and is the adjoint of the Perron-Frobenius operator. The method can be viewed as computing, from the experimental data, the eigenvalues and eigenvectors (low-dimensional modes) of a linear model that approximates the underlying dynamics, even if the dynamics is nonlinear. Since the model is assumed to be linear, the decomposition gives the growth rates and frequencies associated with each mode. If the underlying model is linear, then the DMD method recovers the leading eigenvalues and eigenvectors normally computed using standard solution methods for linear differential equations.

Mathematically, the Koopman operator $\mathbf{A}$ is a linear, time-independent operator $\mathbf{A}$ such that

$$\mathbf{x}_{j+1} = \mathbf{A}\mathbf{x}_j \qquad (5)$$

where j indicates the specific data collection time and $\mathbf{A}$ is the linear operator that maps the data from time t_j to t_{j+1}. The vector $\mathbf{x}_j$ is an N-dimensional vector of the data points collected at time j. The computation of the Koopman operator is at the heart of the DMD methodology. As already stated, the mapping over Δ is linear even though the underlying dynamics that generated $\mathbf{x}_j$ may be nonlinear. It should be noted that this is different than linearizing the dynamics.

To construct the appropriate Koopman operator that best represents the data collected, the matrix $\mathbf{X}_1^{M-1}$ is considered:

$$\mathbf{X}_1^{M-1} = [\mathbf{x}_1 \ \mathbf{x}_2 \ \mathbf{x}_3 \ \cdots \ \mathbf{x}_{M-1}] \, . \tag{6}$$

Making use of (5), this matrix reduces to

$$\mathbf{X}_1^{M-1} = \left[\mathbf{x}_1 \ \mathbf{A}\mathbf{x}_1 \ \mathbf{A}^2\mathbf{x}_1 \ \cdots \ \mathbf{A}^{M-2}\mathbf{x}_1\right] \, . \tag{7}$$

Here is where the DMD method connects to Krylov subspaces and the Arnoldi algorithm. Specifically, the columns of $\mathbf{X}_1^{M-1}$ are each elements in a Krylov space. This matrix attempts to fit the first $M-1$ data collection points using the Koopman operator (matrix) $\mathbf{A}$. In the DMD technique, the final data point $\mathbf{x}_M$ is represented, as best as possible, in terms of this Krylov basis, thus

$$\mathbf{x}_M = \sum_{m=1}^{M-1} b_m \mathbf{x}_m + \mathbf{r} \tag{8}$$

where the b_m are the coefficients of the Krylov space vectors and $\mathbf{r}$ is the residual (or error) that lies outside (orthogonal to) the Krylov space. Ultimately, this best fit to the data using this DMD procedure will be done in an L^2 sense using a pseudo-inverse.

Before proceeding further, it is at this point that the data matrix $\mathbf{X}_1^{M-1}$ in (7) should be considered further. In particular, our dimensionality reduction methods look to take advantage of any low-dimensional structures in the data. To exploit this, the SVD of (7) is computed:

$$\mathbf{X}_1^{M-1} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^* \tag{9}$$

where $*$ denotes the conjugate transpose, $\mathbf{U} \in \mathbb{C}^{N\times K}$, $\mathbf{\Sigma} \in \mathbb{C}^{K\times K}$ and $\mathbf{V} \in \mathbb{C}^{M-1\times K}$. Here K is the reduced SVD's approximation to the rank of $\mathbf{X}_1^{M-1}$. If the data matrix is full rank and the data has no suitable low-dimensional structure, then the DMD method fails immediately. However, if the data matrix can be approximated by a low-rank matrix, then DMD can take advantage of this low dimensional structure to project a future state of the system. Thus once again, the SVD plays the critical role in the methodology.

Armed with the reduction (9) to (7), we can return to the results of the Koopman operator and Krylov basis (8). Specifically, generalizing (5) to its matrix form yields

$$\mathbf{A}\mathbf{X}_1^{M-1} = \mathbf{X}_2^M \, . \tag{10}$$

But by using (8), the right hand side of this equation can be written in the form

$$\mathbf{X}_2^M = \mathbf{X}_1^{M-1}\mathbf{S} + \mathbf{r}e_{M-1}^* \tag{11}$$

where e_{M-1} is the $(M-1)$th unit vector and

$$\mathbf{S} = \begin{bmatrix} 0 & \cdots & & 0 & b_1 \\ 1 & \ddots & & 0 & b_2 \\ 0 & \ddots & \ddots & & \vdots \\ & \ddots & \ddots & 0 & b_{M-2} \\ 0 & \cdots & 0 & 1 & b_{M-1} \end{bmatrix} \, . \tag{12}$$

Recall that the b_j are the unknown coefficients in (8).

The key idea now is the observation that the eigenvalues of $\mathbf{S}$ approximate some of the eigenvalues of the unknown Koopman operator $\mathbf{A}$, making the DMD method similar to the Arnoldi algorithm and its approximations to the Ritz eigenvalues. Schmid [8, 9] showed that rather than computing the matrix $\mathbf{S}$ directly, we can instead compute the *lower-rank* matrix

$$\tilde{\mathbf{S}} = \mathbf{U}^*\mathbf{X}_2^M\mathbf{V}\mathbf{\Sigma}^{-1} \tag{13}$$

which is related to $\mathbf{S}$ via a similarity transformation. Recall that the matrices $\mathbf{U}$, $\mathbf{\Sigma}$ and $\mathbf{V}$ arise from the SVD reduction of $\mathbf{X}_1^{M-1}$ in (9).

Consider then the eigenvalue problem associated with $\tilde{\mathbf{S}}$:

$$\tilde{\mathbf{S}}\mathbf{y}_k = \mu_k \mathbf{y}_k \qquad k = 1, 2, \cdots, K \tag{14}$$

where K is the rank of the approximation we are choosing to make. The eigenvalues μ_k capture the time dynamics of the discrete Koopman map $\mathbf{A}$ as a Δt step is taken forward in time. These eigenvalues and eigenvectors can be related back to the similarity transformed original eigenvalues and eigenvectors of $\mathbf{S}$ in order to construct the DMD modes:

$$\psi_k = \mathbf{U}\mathbf{y}_k \, . \tag{15}$$

With the low-rank approximations of both the eigenvalues and eigenvectors in hand, the projected future solution can be constructed for all time in the future. By first rewriting for convenience $\omega_k = \ln(\mu_k)/\Delta t$ (recall that the Koopman operator time dynamics is linear), then the approximate solution at all future times, $\mathbf{x}_{\text{DMD}}(t)$, is given by

$$\mathbf{x}_{\text{DMD}}(t) = \sum_{k=1}^{K} b_k(0)\psi_k(\mathbf{x})\exp(\omega_k t) = \mathbf{\Psi}\text{diag}(\exp(\omega t)\mathbf{b} \tag{16}$$

where $b_k(0)$ is the initial amplitude of each mode, $\mathbf{\Psi}$ is the matrix whose columns are the eigenvectors ψ_k, $\text{diag}(\omega t)$ is a diagonal matrix whose entries are the eigenvalues $\exp(\omega_k t)$, and $\mathbf{b}$ is a vector of the coefficients b_k.

It only remains to compute the initial coefficient values $b_k(0)$. If we consider the initial snapshot ($\mathbf{x}_1$) at time zero, let's say, then (16) gives $\mathbf{x}_1 = \mathbf{\Psi}\mathbf{b}$. This generically is not a square matrix so that its solution

$$\mathbf{b} = \mathbf{\Psi}^+\mathbf{x}_1 \tag{17}$$

can be found using a pseudo-inverse. Indeed, $\mathbf{\Psi}^+$ denotes the Moore-Penrose pseudo-inverse that can be accessed in MATLAB via the **pinv** command. As already discussed in the compressive sensing section, the pseudo-inverse is equivalent to finding the best solution $\mathbf{b}$ the in the least-squares (best fit) sense. This is equivalent to how DMD modes were derived originally.

Overall then, the DMD algorithm presented here takes advantage of low dimensionality in the data in order to make a low-rank approximation of the linear mapping that best approximates the nonlinear dynamics of the data collected for the system. Once this is done, a prediction of the future state of the system is achieved for all time. Unlike the POD method, which requires solving a low-rank set of dynamical quantities to predict the future state, no additional work is required for the future state prediction outside of plugging in the desired future time into (16). Thus the advantages of DMD revolve around the fact that (i) no equations are needed, and (ii) the future state is known for all time (of course, provided the DMD approximation holds).

The algorithm is as follows:

(i) Sample data at N prescribed locations M times. The data snapshots should be evenly spaced in time by a fixed Δt. This gives the data matrix $\mathbf{X}$.

(ii) From the data matrix $\mathbf{X}$, construct the two sub-matrices $\mathbf{X}_1^{M-1}$ and $\mathbf{X}_2^{M}$.

(iii) Compute the SVD decomposition of $\mathbf{X}_1^{M-1}$.

(iv) The matrix $\tilde{\mathbf{S}}$ can then be computed and its eigenvalues and eigenvectors found.

(v) Project the initial state of the system onto the DMD modes using the pseudo-inverse.

(vi) Compute the solution at any future time using the DMD modes along with their projection to the initial conditions and the time dynamics computed using the eigenvalue of $\tilde{\mathbf{S}}$.

3. MULTI-RESOLUTION ANALYSIS

The MRDMD is inspired by the observation that the slow- and fast-modes can be separated for such applications as foreground/background subtraction in videos feeds [14]. The MRDMD recursively removes low-frequency content from a given collection of snapshots. Typically, the number of snapshots M are chosen so that the DMD modes provide an approximately full rank approximation of the dynamics observed. Thus the M is chosen so that all high- and low-frequency content is present. In the MRDMD, M is originally chosen in the same way so that an approximate full rank approximation can be accomplished. However, from this initial pass through the data, the slowest m_1 modes are removed and DMD is once again performed with now only $M/2$ snapshots. Again the slowest m_2 modes are removed and the algorithm is continued until a desired termination.

Mathematically, the MRDMD separates the DMD approximate solution (16) in the first pass as follows:

$$\begin{aligned}\mathbf{x}_{\text{DMD}}(t) &= \sum_{k=1}^{M} b_k(0)\psi_k^{(1)}(\mathbf{x})\exp(\omega_k t) \qquad (18)\\ =\sum_{k=1}^{m_1} b_k(0)\psi_k^{(1)}(\mathbf{x})\exp(\omega_k t) &+ \sum_{k=m_1}^{M} b_k(0)\psi_k^{(1)}(\mathbf{x})\exp(\omega_k t)\\ \text{(slow modes)} &\qquad\qquad \text{(fast modes)}\end{aligned}$$

where the $\psi_k^{(1)}(\mathbf{x})$ represent the DMD modes computed from the full M snapshots.

The first sum in this expression (19) represents the slow-mode dynamics whereas the second some is everything else. Thus the second sum can be computed to yield the matrix:

$$\mathbf{X}_{M/2} = \sum_{k=m_1}^{M} b_k(0)\psi_k^{(1)}(\mathbf{x})\exp(\omega_k t)\,. \qquad (19)$$

The DMD analysis outlined in the previous section can now be performed once again on the data matrix $\mathbf{X}_{M/2}$. However, the matrix $\mathbf{X}_{M/2}$ is now separated into two matrices

$$\mathbf{X}_{M/2} = \mathbf{X}_{M/2}^{(1)} + \mathbf{X}_{M/2}^{(2)} \qquad (20)$$

where the first matrix contains the first $M/2$ snapshots and the second matrix contains the remaining $M/2$ snapshots. The m_2 slow-DMD modes at this level are given by $\psi_k^{(2)}$, where they are computed separately in the first of second interval of snapshots.

The iteration process works by recursively removing slow frequency components and building the new matrices $\mathbf{X}_{M/2}$, $\mathbf{X}_{M/4}, \mathbf{X}_{M/8}, \cdots$ until a desired/prescribed multi-resolution decomposition has been achieved. The approximate DMD solution can then be constructed as follows:

$$\begin{aligned}\mathbf{x}_{\text{DMD}}(t) &= \sum_{k=1}^{m_1} b_k^{(1)}\psi_k^{(1)}(\mathbf{x})\exp(\omega_k^{(1)} t)\\ &+ \sum_{k=1}^{m_2} b_k^{(2)}\psi_k^{(2)}(\mathbf{x})\exp(\omega_k^{(2)} t) \qquad (21)\\ &+ \sum_{k=1}^{m_3} b_k^{(3)}\psi_k^{(3)}(\mathbf{x})\exp(\omega_k^{(3)} t) + \cdots\end{aligned}$$

where the $\psi_k^{(k)}$ and $\omega_k^{(k)}$ are the DMD modes and DMD eigenvalues at the kth level of decomposition, the $b_k^{(k)}$ are the initial projections of the data unto the time interval of

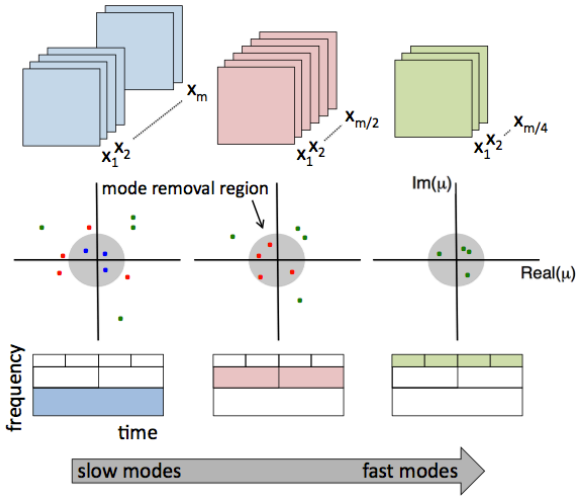

Fig. 2. Representation of the multi-resolution dynamic mode decomposition where successive sampling of the data, initially with M snapshots and decreasing by a factor of two at each resolution level, is shown (top figures). The DMD spectrum is shown in the middle panel where there are m_1 (blue dots) slow-dynamic modes at the slowest level, m_2 (red) modes at the next level and m_3 (green) modes at the fastest time-scale shown. The shaded region represents the modes that are removed at that level. The bottom panels shows the wavelet-like time-frequency decomposition of the data color coded with the snapshots and DMD spectral representations.

Fig. 3. Demonstration of the foreground/background separation of videos using the DMD algorithm. The example problem is motivated by a surveillance problem which tries to identify moving cars and pedestrians from stationary/parked cars in a video feed.

interest, and the m_k are the number of slow-modes retained at each level. The advantage of this method is readily apparent: different spatial-temporal DMD modes are used to represent key multi-resolution features. Thus there is not a single set of modes that dominates the SVD decomposition and potentially marginalizes features at other time scales.

Figure 2 illustrate the multi-resolution DMD process pictorially. In the figure, a three-level decomposition is performed with the slowest scale represented in blue (eigenvalues and snapshots), the mid-scale in red and the fast scale in green. The connection to multi-resolution wavelet analysis is also evident from the bottom panels as one can see that the MRDMD method successively pulls out time-frequency information in a principled way.

4. VIDEO FOREGROUND AND BACKGROUND SEPARATION

Using the Advanced Video and Signal based Surveillance (AVSS) Datasets, specifically the "Parked Vehicle - Hard" video, the DMD separation procedure can applied. The original videos are converted to grayscale and down-sampled in pixel resolution to $n = 120 \times 96 = 11520$, in order to make the computational memory requirements manageable for personal computers. Also, the introductory preambles to the surveillance videos, which constitute the first 351 frames of each video, are removed because they are irrelevant for the following illustrations. The video streams are broken into segments of $m = 30$ frames each. Frame numbers 500, 1000, and 2000 of the entire video stream are depicted in Fig. 3, along with the separation results for easy comparison. For enhanced contrast and better visibility, the sparse results are artificially brightened by a factor of 10. The AVSS "Parked Vehicle" surveillance video, which generally shows various vehicles traveling along a road, with a traffic light (not visible) and a crosswalk (visible) near the bottom of the frame, and with an occasional vehicle parking along side the road. Sometimes, In the distance, moving vehicles become difficult to perceive with the naked-eye, due to limitations of the pixel resolution. For all three frames, the DMD method seems to eliminate more spurious pixels in its sparse results that may pertain to the background when compared to the RPCA's sparse results. A good background/foreground separation such as this can form the basis of surveillance techniques which are real-time using limited computation.

5. CONCLUSIONS

Overall it has been demonstrated that the method of dynamic mode decomposition, typically used for evaluating the dynamics of complex systems, can be used for background/foreground separation in videos with visually appealing results and excellent computational efficiency. The separation results produced by the DMD method are on par with the quality of separation achieved with the RPCA method for realistic video scenarios. However, the results are achieved orders of magnitude faster. Indeed, we demonstrate that DMD is viable as a real-time solution to foreground/background video separation tasks even with laptop-level computing platforms. As with any separation method, including RPCA and DMD, the burden of working with too much data, i.e. high-resolution images and/or many frames per video segment, can be problematic because of reduced computational speeds and limited memory sizes. Nonetheless, the DMD algorithm has shown itself to be robust and efficient enough to produce attractive results in times well below the normal frame acquisition rate of most cameras, allowing for higher pixel resolutions and video segment sizes to be used. For real-time video applications, it makes sense to break the continuous video stream into segments large enough to ensure that there is enough information to complete an adequate background/foreground separation, but small enough to keep the processing times smaller than the data acquisition times. Additionally, moving objects that turn, stop, and/or accelerate are better handled by the DMD procedure as individual actions, than in one large video segment. The additional innovation around MRDMD also allows one to separate objects that are dynamic on different time scales. It is a principled technique by which such data surveillance can achieved in real-time using well established ideas of multi-resolution analysis.

6. REFERENCES

[1] L. Li, W. Huang, I. Gu, and Q. Tian. *Statistical Modeling of Complex Backgrounds for Foreground Object Detection. IEEE Transactions on Image Processing*, 13(11):1459–1472, 2004.

[2] Y. Tian, M. Lu, and A. Hampapur. *Robust and Efficient Foreground Analysis for Real-Time Video Surveillance.* In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005.*, volume 1, pages 1182–1187, 2005.

[3] L. Maddalena and A. Petrosino. *A Self-Organizing Approach to Background Subtraction for Visual Surveillance Applications. IEEE Transactions on Image Processing*, 17(7):1168–1177, 2008.

[4] J. He, L. Balzano, and A. Szlam. *Incremental Gradient on the Grassmannian for Online Foreground and Background Separation in Subsampled Video.* In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1568–1575, 2012.

[5] E. Candès, X. Li, Y. Ma, and J. Wright. *Robust Principal Component Analysis? Computing Research Repository*, abs/0912.3599, 2009.

[6] Y. Benezeth, P.-M. Jodoin, B. Emile, H. Laurent, and C. Rosenberger. *Comparative Study of Background Subtraction Algorithms. Journal of Electronic Imaging 19 (2010)*, 19(3):033003, 2010.

[7] F. De la Torre and M. Black. *A Framework for Robust Subspace Learning. International Journal of Computer Vision*, 54(1-3):117–142, 2003.

[8] P. Schmid. *Dynamic mode decomposition of numerical and experimental data. Journal of Fluid Mechanics*, 656:5–28, 2010.

[9] P. Schmid, L. Li, M. Juniper, and O. Pust. *Applications of the dynamic mode decomposition. Theoretical and Computational Fluid Dynamics*, 25(1-4):249–259, 2011.

[10] K. Chen, J. Tu, and C. Rowley. *Variants of Dynamic Mode Decomposition: Boundary Condition, Koopman, and Fourier Analyses. Journal of Nonlinear Science*, 22(6):887–915, 2012.

[11] C. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. Henningson. *Spectral analysis of nonlinear flows. Journal of Fluid Mechanics*, 641:115–127, 2009.

[12] J. Tu, D. Luchtenberg, C. Rowley, S. Brunton, and J. N. Kutz. *Generalizing dynamic mode decomposition to a larger class of datasets. Journal of Computational Dynamics*, 2013. (submitted).

[13] J. N. Kutz. *Data-driven modeling and scientific computing: Methods for Integrating Dynamics of Complex Systems and Big Data.* Oxford Press, 2013.

[14] J. Grosek and J. N. Kutz. *Dynamic Mode Decomposition for Real-Time Background/Foreground Separation in Video. arXiv*, page arXiv:1404.7592, 2014.