Efficient and Robust Background Modeling with Dynamic Mode Decomposition

Tim Krake^{1,3} · Andrés Bruhn² · Bernhard Eberhardt³ · Daniel Weiskopf¹

Received: 18 June 2021 / Accepted: 20 January 2022 / Published online: 24 February 2022
© The Author(s) 2022

Abstract

A large number of modern video background modeling algorithms deal with computational costly minimization problems that often need parameter adjustments. While in most cases spatial and temporal constraints are added artificially to the minimization process, our approach is to exploit Dynamic Mode Decomposition (DMD), a spectral decomposition technique that naturally extracts spatio-temporal patterns from data. Applied to video data, DMD can compute background models. However, the original DMD algorithm for background modeling is neither efficient nor robust. In this paper, we present an equivalent reformulation with constraints leading to a more suitable decomposition into fore- and background. Due to the reformulation, which uses sparse and low-dimensional structures, an efficient and robust algorithm is derived that computes accurate background models. Moreover, we show how our approach can be extended to RGB data, data with periodic parts, and streaming data enabling a versatile use.

Keywords Dynamic mode decomposition · Spectral decomposition · Background modeling · Foreground detection

1 Introduction

Modern video background modeling algorithms are often inherently based on matrix decompositions [6]. Besides pre- and post-processing, the core of these algorithms is to explicitly or implicitly decompose the data into a low-rank matrix characterizing the background and a sparse matrix that shows the foreground. To perform such a decomposition, a computational costly minimization problem with param-

eter adjustment is typically established that neglects spatial and temporal coherence at first (due to the use of simple matrix norms, seminorms, and quasinorms in the loss function). Since this formulation may fail to produce adequate background models for more complex video sequences, spatial and/or temporal constraints are taken into account to meet the requirements. Another more recent approach is to integrate neural networks [1] or related concepts into the process of background modeling, which leads to supervised methods in general. In this paper, however, we neither focus on a minimization-based background modeling approach as described above nor on learning-based concepts. Instead, we follow a different approach and exploit Dynamic Mode Decomposition (DMD), a versatile spectral decomposition technique that naturally extracts spatio-temporal patterns from data.

DMD was originally established in the fluid dynamics community [30] and has evolved to an effective analysis tool. Numerically, DMD produces the following triples: DMD modes, amplitudes, and eigenvalues. For time-dependent data, the modes represent the spatial contribution and the amplitudes specify their impact. Each associated eigenvalue characterizes the temporal development. According to this interpretation, Grosek and Kutz [14] were the first to model the background in video by DMD. Compared to state-of-

✉ Tim Krake
Tim.Krake@visus.uni-stuttgart.de;
Tim.Krake@hdm-stuttgart.de

Andrés Bruhn
Andres.Bruhn@vis.uni-stuttgart.de

Bernhard Eberhardt
Bernhard.Eberhardt@hdm-stuttgart.de

Daniel Weiskopf
Daniel.Weiskopf@visus.uni-stuttgart.de

¹ Visualization Research Center (VISUS), University of Stuttgart, Allmandring 19, 70569 Stuttgart, Germany

² Institute for Visualisation and Interactive Systems (VIS), University of Stuttgart, Universitätsstraße 38, 70569 Stuttgart, Germany

³ Hochschule der Medien, Nobelstraße 10, 70569 Stuttgart, Germany

the-art algorithms, the approach suffers in terms of accuracy, efficiency, and robustness. Some publications address the high computational costs of DMD by efficient subsampling strategies or related concepts, but these techniques struggle even more with problems regarding accuracy and robustness.

Instead of applying subsampling techniques, we present an equivalent reformulation of DMD with constraints that enforces the existence of background components. Due to the reformulation, which uses sparse and low-dimensional structures, extremely efficient algorithms (e.g., the power method) can be leveraged for the computation. The contribution of this paper can be summarized as follows:

- We elaborate on the drawbacks of previous methods and thus motivate our approach.
- An equivalent reformulation of DMD with constraints is presented, leading to a suitable decomposition into foreground and background.
- We derive an efficient and robust algorithm that computes accurate background models.
- The approach is extended to RGB data, data with periodic background parts, and streaming data.

The remainder of this paper is organized as follows: After clarifying related work in the subsequent section, Sect. 3 reviews the traditional DMD approach for foreground–background separation. Section 4 presents our approach including a motivation, new formulation, the associated algorithm, extensions, and implementation details. Section 5 provides a qualitative and quantitative evaluation in terms of accuracy, efficiency, and robustness. Finally, a conclusion is drawn in Sect. 6.

2 Related Work

There are numerous background modeling algorithms developed in the literature [5], and they can be classified as unsupervised [26,29,37], supervised [1,4,7,23,24], and semi-supervised methods [12,13]. Supervised methods allow for more accurate results than unsupervised methods, especially

in complex scenarios, but they need typically large training data. Semi-supervised approaches achieve comparably accurate results with less labeled training data; however, they are prone to supervision aspects as well. Since our approach does not rely on training processes or related concepts, it is an unsupervised method. Such methods were published in large numbers and varieties over the past decade. Depending on the tasks, papers [11,34,39] classify them differently. To our knowledge, the most recent review for unsupervised background modeling algorithms was written by Bouwmans et al. [6] in combination with the LRSlibrary [32]. Using the approach of decomposing video data into a low-rank and sparse matrix, they classify algorithms into the following categories: robust principal component analysis (RPCA), matrix completion (MC), non-negative matrix factorization (NMF), subspace tracking (ST), low-rank recovery (LRR), three-term decomposition (TTD), and tensor decompositions. Bouwmans et al. [6] put DMD into the RPCA category. However, DMD substantially differs from conventional RPCA approaches, e.g., the structure of the established minimization problems is completely different. We thus compare our approach only with DMD-based approaches in the following.

The first background modeling approach with DMD was proposed by Grosek and Kutz [14]. An extension to RGB video was presented by Tirunagari [35]. However, alternative state-of-the-art background modeling algorithms achieve better results with substantially less computational costs. Kutz et al. [21] later proposed using multi-resolution DMD for background modeling. It integrates multi-resolution analysis techniques into DMD, which enables the identification of multi-time scale features and, therefore, may lead to better results. Nevertheless, computational costs are even higher. The issue of efficiency was addressed by Erichson and Donovan [10] and slightly later by Erichson et al. [9] and Pendergrass et al. [28]. All these approaches use efficient sampling strategies, either based on a randomized singular value decomposition (randomized DMD) or on compressed data (compressed DMD). Whereas these subsampling techniques show promising efficiency, the background model suffers from the subsampling process in terms of accuracy

Fig. 1 The traditional DMD approach for background modeling: After vectorizing the $m + 1$ video frames into n -dimensional data, the DMD modes, eigenvalues, and amplitudes are computed via a low-dimensional representation. Finally, the background model is extracted

Algorithm 1 Exact Dynamic Mode Decomposition

```

1: function EXDMD( $x_0, \dots, x_m$ )
2:    $X = [x_0 \dots x_{m-1}]$ ,  $Y = [x_1 \dots x_m]$ 
3:   Reduced SVD:  $X = U \Sigma V^*$  with  $\text{rank}(X) = r$ 
4:   Calculate  $S_{\text{DMD}} = U^* Y V \Sigma^{-1}$ 
5:   Calculate  $\lambda_1, \dots, \lambda_r$  and  $v_1, \dots, v_r$  of  $S_{\text{DMD}}$ 
6:   for  $\lambda_i \neq 0$  do
7:      $\vartheta_i = \frac{1}{\lambda_i} Y V \Sigma^{-1} v_i$ 
8:   end for
9:    $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_{r_0})$  with  $\lambda_1, \lambda_2, \dots, \lambda_{r_0} \neq 0$ 
10:   $\Theta = [\vartheta_1 \ \vartheta_2 \ \dots \ \vartheta_{r_0}]$ 
11:  Calculate  $a = \text{lsqr}(\Theta, x_0)$  with  $a = (a_1, \dots, a_{r_0})$ 
12:  return  $\lambda_j, \vartheta_j, a_j$ 
13: end function

```

and robustness. Instead of applying subsampling techniques, we use an equivalent reformulation of DMD with constraints that allows efficient and robust computation and leads to a more suitable decomposition into fore- and background.

Recently, Haq et al. [16] also presented an efficient DMD-based method for background modeling that uses dictionary learning as subsampling strategy. As the dictionary needs to be trained, it substantially differs from our approach.

Finally, there are also other DMD-based techniques for videos, e.g., video shot detection [3] or saliency detection [27]. However, those approaches have other objectives and technical contributions than our paper.

3 Theoretical Foundation

This section provides the basics of the foreground–background separation with DMD, including the general theory of DMD and its traditional application to video data (see Fig. 1).

3.1 Dynamic Mode Decomposition

Before formulating the DMD background modeling algorithm, we summarize the principles of DMD in general [19, 20]. These aspects help understand foreground–background separation and, thus, our approach.

Let us consider data $x_0, \dots, x_m \in \mathbb{R}^n$ sampled uniformly in time. Typically, the data dimension n is considerably greater than m (compare Fig. 1, left box). DMD basically computes eigenvalues and eigenvectors of a high-dimensional matrix $A_{\text{DMD}} \in \mathbb{R}^{n \times n}$ (characterizing the dynamics of the data) via a low-dimensional representation S_{DMD} (see Fig. 1, middle box). The matrix A_{DMD} is derived from the minimization problem

$$\min \|AX - Y\|_F^2, \quad (1)$$

where $X = [x_0 \dots x_{m-1}]$, $Y = [x_1 \dots x_m]$. In general, the minimization problem has infinite solutions (as we will prove

later). DMD therefore aims for the minimum-norm solution, which is explicitly given by

$$A_{\text{DMD}} = YX^+, \quad (2)$$

where X^+ is the Moore–Penrose pseudoinverse of X . To obtain the eigenvalues and eigenvectors of A_{DMD} , the low-dimensional representation S_{DMD} is computed via the thin singular value decomposition (SVD) of X , i.e., $X = U \Sigma V^*$ with $r = \text{rank}(X)$, and the following reformulation:

$$S_{\text{DMD}} = U^* A_{\text{DMD}} U = U^* Y V \Sigma^{-1} U^* U = U^* Y V \Sigma^{-1}. \quad (3)$$

Eq. 3 is the core of DMD, forming the starting point of Algorithm 1 (Lines 1–4). In the next steps, the (DMD) eigenvalues $\lambda_j \in \mathbb{C}$ and eigenvectors $v_j \in \mathbb{C}^r$ of S_{DMD} are computed (line 5). Finally, all eigenvectors with nonzero eigenvalues are transformed into modes $\vartheta_j = \frac{1}{\lambda_j} Y V \Sigma^{-1} v_j \in \mathbb{C}^n$ (Lines 6–8) and the corresponding amplitudes $a_j \in \mathbb{C}$ are computed by $\min \|\Theta a - x_0\|$ (Lines 9–11). Algorithm 1 results in the following decomposition of the data:

$$x_k \approx \sum_{j=1}^{r_0} \lambda_j^k a_j \vartheta_j. \quad (4)$$

This approximation property is crucial for the interpretation of DMD triples $(\vartheta_j, \lambda_j, a_j)$, in particular, concerning the foreground–background separation approach.

3.2 Background Modeling with DMD

To compute a background model with DMD, the DMD components have to be interpreted in terms of video data. An overview of the traditional DMD background modeling is illustrated in Fig. 1. After applying DMD to the vectorized video frames, DMD produces modes $\vartheta_j \in \mathbb{C}^n$, eigenvalues $\lambda_j \in \mathbb{C}$, and amplitudes $a_j \in \mathbb{C}$ with a computational complexity of $\mathcal{O}(nm^2)$ in general as $n \gg m$. The entries of a DMD mode ϑ_j correspond to grayscale pixel values at specific pixel locations. According to Eq. 4, the associated eigenvalue λ_j characterizes the temporal development. To extract a background model, Grosek and Kutz [14] propose selecting DMD triples that vary over time extremely slowly or not even at all. These triples are characterized by eigenvalues that satisfy $\lambda_j \approx 1$ or, more precisely, by the index set

$$I_{\mathbb{1}} = \{j \in \{1, \dots, r_0\} ; |\lambda_j - 1| < \delta\}$$

for a given threshold δ (see Fig. 2, top). Therefore, the video frames can be separated by DMD into fore- and background:

Fig. 2 Comparison of the traditional approach with ours: The traditional approach uses eigenvalues of the matrix S_{DMD} for the background that are only approximately 1. Therefore, fore- and background components can be mixed up as the extracted background at a certain time shows. In contrast, our approach uses a projection that enforces the existence of an eigenvalue $\lambda_{\mathbb{I}} = 1$, which leads to a more suitable decomposition. To further improve the background model, the extended method should be consulted, where multiple eigenvalues are used

$$x_k \approx x_k^{\text{FG}} + x_k^{\text{BG}} = \sum_{j \notin I_{\mathbb{I}}} \lambda_j^k a_j \vartheta_j + \sum_{j \in I_{\mathbb{I}}} \lambda_j^k a_j \vartheta_j. \quad (5)$$

Since the eigenvalues λ_j , amplitudes a_j , and modes ϑ_j are complex in general, the sums are not real-valued. Whereas Grosek and Kutz [14] use the absolute value as well as a rearrangement of negative values, Erichson et al. [9] recommend the use of the real part. In this paper, we follow the technique of Grosek and Kutz.

4 Proposed Approach

In this section, we present our DMD background modeling approach. After a short motivation, the theory and the algorithm (based on grayscaled data) are explained in detail. We then extend our approach to RGB data, data with periodic parts, and streaming data, which enables the usage in video surveillance. Finally, the implementation is described.

4.1 Motivation

Leading background modeling algorithms are characterized by accuracy, robustness, and efficiency. While DMD is capable of producing accurate background models in general, the traditional DMD background modeling algorithm is neither robust nor efficient. This fact can be seen in Fig. 1, where the traditional approach fails to produce an accurate background model. We additionally highlight the computational costs of each step of DMD. Since both an SVD and an eigenvalue decomposition is needed, DMD has a computational complexity of $\mathcal{O}(nm^2)$ in general as $n \gg m$.

Even though the improvements via subsampling strategies boost efficiency at the cost of accuracy, they do not solve the following underlying problem: In Fig. 2 top, the background model of the traditional approach is shown at an earlier frame. It can be observed that a pair of complex-conjugated eigenvalues close to the value 1 is detected. According to the traditional approach, the two corresponding DMD triples $(\vartheta_j, a_j, \lambda_j)$ should be selected. If they are not close enough at the value 1, no components are selected due to thresholding and no background can be computed, i.e., the algorithm is not robust. Otherwise, this small inaccuracy causes numerical instability and a mixing of fore- and background parts as shown in Fig. 2. These issues may be even worse for subsampled data.

Consequently, DMD has to be modified such that components either clearly belong to the fore- or background. Since DMD only uses one or two triples (often occurring in complex-conjugated pairs) for the background model [20], we propose enforcing the existence of an eigenvalue $\lambda_{\mathbb{I}} = 1$ by a DMD-consistent projection. Moreover, we also recommend using the efficient power method to compute the relevant DMD components for an adequate background model. We can do this because our experiments have shown that all other eigenvalues are inside the unit circle due to the fact that foreground objects appear and disappear over time. This leads to a more suitable decomposition into fore- and background as shown in Fig. 2 that is produced in an efficient manner.

4.2 Theory

As mentioned before, we want to enforce the existence of an eigenvalue $\lambda_{\mathbb{I}} = 1$ to obtain a suitable decomposition into fore- and background. To understand the general idea of our approach, we shortly summarize the steps in the following:

- (§1) Prove the existence of infinite solutions of the minimization problem 1 (see Proposition 1).
- (§2) Formulate a new minimization problem (see Eq. 6) and find a more accessible representation (see Theorem 1, Theorem 2, and Eq. 11).
- (§3) Find a feasible solution to the final minimization problem (see Theorem 3).

(§1) If the data $x_0, \dots, x_m \in \mathbb{R}^n$ are linear independent (which is the case for real video footage in general), we first note that the high-dimensional minimization problem 1 has infinite solutions that solve the equation exactly, i.e., $AX = Y$.

Proposition 1 For linearly independent data $x_0, \dots, x_m \in \mathbb{R}^n$ with $Z = [x_0 \dots x_m] \in \mathbb{R}^{n \times m+1}$, the minimization problem 1 has infinite solutions that solve the equation exactly,

i.e., $AX = Y$. In particular, every combination of $m + 1$ distinct complex numbers $z_0, \dots, z_m \in \mathbb{C}$ defines a solution A_z in the form of its eigenvalues.

Proof Let us consider distinct complex numbers $z_0, \dots, z_m \in \mathbb{C}$ with $\Lambda_z = \text{diag}(z_0, \dots, z_m)$ as well as the respective Vandermonde matrix

$$V(z) = \begin{pmatrix} z_0^0 & \dots & z_0^m \\ \vdots & \ddots & \vdots \\ z_m^0 & \dots & z_m^m \end{pmatrix} \in \mathbb{C}^{(m+1) \times (m+1)}.$$

Defining $\Xi_z = ZV(z)^{-1}$ (the columns of the matrix are basically scaled modes), the matrix $A_z = \Xi_z \Lambda_z \Xi_z^+$ solves the minimization problem 1 exactly:

$$\begin{aligned} A_z X &= \Xi_z \Lambda_z \Xi_z^+ X \\ &= ZV(z)^{-1} \Lambda_z V(z) Z^+ X \\ &= ZV(z)^{-1} \Lambda_z V(z) \begin{bmatrix} I_m \\ 0^T \end{bmatrix} \\ &= ZV(z)^{-1} \Lambda_z \begin{pmatrix} z_0^0 & \dots & z_0^{m-1} \\ \vdots & \ddots & \vdots \\ z_m^0 & \dots & z_m^{m-1} \end{pmatrix} \\ &= ZV(z)^{-1} \begin{pmatrix} z_0^1 & \dots & z_0^m \\ \vdots & \ddots & \vdots \\ z_m^1 & \dots & z_m^m \end{pmatrix} = Z \begin{bmatrix} 0^T \\ I_m \end{bmatrix} = Y \end{aligned}$$

where we used that $\Xi_z^+ = V(z)Z^+$ (as the columns of the matrix Z are linearly independent). Since $z_0, \dots, z_m$ are chosen arbitrarily, the assertion is proven. $\square$

The above theorem states that every distinct combination of temporal components $\lambda_0, \dots, \lambda_m$ (which are eigenvalues of a respective matrix A) are actually possible outcomes. However, since DMD uses the minimum-norm solution $A_{\text{DMD}} = YX^+$, the DMD eigenvalues fit better to the data than a random solution. The existence of adequate eigenvalues for background modeling is not guaranteed nonetheless. (§2) Hence, the naive approach is to find a matrix $A \in \mathbb{R}^{n \times n}$ that solves the minimization problem 1 and has an eigenvalue $\lambda_{\mathbb{I}} = 1$:

$$\begin{aligned} \min \quad & \|A - A_{\text{DMD}}\|_F^2 \\ \text{s.t.} \quad & AX = Y \\ & \lambda_{\mathbb{I}} \text{ is eigenvalue of } A. \end{aligned} \quad (6)$$

However, this formulation cannot be solved practically due to the high-dimensionality of the problem and the eigenvalue equation in the constraints.

To formulate an adequate constrained minimization problem, we use the low-dimensional representation C of a matrix

A with respect to the data matrix $Z = [x_0 \dots x_m] \in \mathbb{R}^{n \times (m+1)}$:

$$C = Z^+ A Z \quad (7)$$

This matrix is a companion matrix and completely characterizes the data-relevant spectral properties of A . In addition, the retransformed matrix $ZCZ^+ \in \mathbb{R}^{n \times n}$ inherits many algebraic properties, in particular, the minimization property, as the following theorem shows.

Theorem 1 For linearly independent data $x_0, \dots, x_m \in \mathbb{R}^n$ with $Z = [x_0 \dots x_m] \in \mathbb{R}^{n \times (m+1)}$, the low-dimensional representation $C = Z^+ A Z \in \mathbb{R}^{(m+1) \times (m+1)}$ to a matrix $A \in \mathbb{R}^{n \times n}$ satisfying $AX = Y$ has the following properties:

(a) The low-dimensional representation C is a companion matrix

$$C = \begin{bmatrix} 0^T & \tilde{c} \\ I_m & c \end{bmatrix} = \begin{bmatrix} 0^T & c_0 \\ I_m & c \end{bmatrix} \quad (8)$$

and the related vector $\tilde{c}$ satisfies

$$\tilde{c} = \begin{pmatrix} c_0 \\ c \end{pmatrix} = Z^+ A x_m \in \mathbb{R}^{m+1}. \quad (9)$$

(b) An eigenvalue λ of the matrix C with eigenvector $\tilde{v}$ is also an eigenvalue of A with corresponding eigenvector $v = Z\tilde{v}$, if $Av \in \text{span}(x_0, \dots, x_m)$.

(c) An eigenvalue $\lambda \neq 0$ of the matrix A with eigenvector v is also an eigenvalue of C with corresponding eigenvector $\tilde{v} = Z^+ v$, if $v \in \text{span}(x_0, \dots, x_m)$.

(d) The retransformed matrix $ZCZ^+ \in \mathbb{R}^{n \times n}$ satisfies the equation $(ZCZ^+)X = Y$.

Proof (a) The following calculation shows that C is a companion matrix:

$$C = Z^+ A Z = Z^+ [Y \ A x_m] = \begin{bmatrix} 0^T & c_0 \\ I_m & c \end{bmatrix},$$

where $c_0 \in \mathbb{R}$ and $c \in \mathbb{R}^m$ are values that satisfy

$$\tilde{c} = \begin{pmatrix} c_0 \\ c \end{pmatrix} = Z^+ A x_m \in \mathbb{R}^{m+1}.$$

(b) Let λ be an eigenvalue of the matrix C with eigenvector $\tilde{v}$. Then, the nonzero vector $v = Z\tilde{v}$ (as Z has linearly independent columns) satisfies the following equation

$$Av = ZZ^+ Av = ZZ^+ A Z \tilde{v} = ZC \tilde{v} = \lambda Z \tilde{v} = \lambda v,$$

where we have used that ZZ^+ is the projection onto the image of Z .

(c) Let λ be an eigenvalue of the matrix A with eigenvector v . Then, the vector $\tilde{v} = Z^+v$ satisfies the following equation

$$C\tilde{v} = CZ^+v = Z^+AZZ^+v = Z^+Av = \lambda Z^+v = \lambda\tilde{v}.$$

To show that $\tilde{v}$ is an eigenvector, we assume that $\tilde{v} = 0$. This implies $0 = AZ\tilde{v}$, which results in $0 = AZZ^+v = Av = \lambda v$. Since $v \neq 0$, the eigenvalue has to be zero, i.e., $\lambda = 0$. However, this contradicts the assumption. Therefore, $\tilde{v} \neq 0$.
(d) Using the companion matrix structure and the linearly independent columns of the matrix Z , the following calculation shows that the retransformed matrix ZCZ^+ solves the first constraint:

$$(ZCZ^+)X = ZC \begin{bmatrix} I_m \\ 0^T \end{bmatrix} = Z \begin{bmatrix} 0^T & c_0 \\ I_m & c \end{bmatrix} \begin{bmatrix} I_m \\ 0^T \end{bmatrix} = Z \begin{bmatrix} 0^T \\ I_m \end{bmatrix} = Y.$$

□

With the help of Theorem 1, every solution A to the minimization problem 1 can be represented by an appropriate companion matrix C that naturally satisfies the first constraint in minimization problem 6. To reformulate the entire constrained minimization problem 6, the low-dimensional representation of $A_{\text{DMD}} = YX^+$ has to be calculated as well, which will be done in the following theorem.

Theorem 2 For linearly independent data $x_0, \dots, x_m \in \mathbb{R}^n$ with $Z = [x_0 \dots x_m] \in \mathbb{R}^{n \times m+1}$, the low-dimensional representation $C_{\text{DMD}} = Z^+A_{\text{DMD}}Z \in \mathbb{R}^{m+1 \times m+1}$ to the matrix $A_{\text{DMD}} = YX^+ \in \mathbb{R}^{n \times n}$ has the following properties:

(a) The low-dimensional representation C_{DMD} is a companion matrix with $\text{rank}(C) = m$ and the related vector $\tilde{c}_{\text{DMD}}$ satisfies

$$\tilde{c}_{\text{DMD}} = \begin{pmatrix} c_0 \\ c_{\text{DMD}} \end{pmatrix} = \begin{pmatrix} 0 \\ X^+x_m \end{pmatrix}. \quad (10)$$

- (b) An eigenvalue λ of the matrix C_{DMD} with eigenvector $\tilde{v}$ is also an eigenvalue of A_{DMD} with corresponding eigenvector $v = Z\tilde{v}$.
(c) An eigenvalue $\lambda \neq 0$ of the matrix A_{DMD} with eigenvector v is also an eigenvalue of C_{DMD} with corresponding eigenvector $\tilde{v} = Z^+v$.
(d) The retransformed matrix $ZC_{\text{DMD}}Z^+$ is equal to A_{DMD} , i.e., $ZC_{\text{DMD}}Z^+ = A_{\text{DMD}}$ (trivially satisfying $(ZC_{\text{DMD}}Z^+)X = Y$).

Proof (a) According to Theorem 1, the matrix C_{DMD} is a companion matrix

$$C_{\text{DMD}} = \begin{bmatrix} 0^T & c_0 \\ I_m & c_{\text{DMD}} \end{bmatrix},$$

where the corresponding vector $\tilde{c}_{\text{DMD}}$ is given by

$$\begin{aligned} \tilde{c}_{\text{DMD}} &= \begin{pmatrix} c_0 \\ c_{\text{DMD}} \end{pmatrix} = Z^+A_{\text{DMD}}x_m = Z^+YX^+x_m \\ &= \begin{bmatrix} 0^T \\ I_m \end{bmatrix} X^+x_m = \begin{pmatrix} 0 \\ X^+x_m \end{pmatrix}. \end{aligned}$$

Since $c_0 = 0$, the matrix C_{DMD} has $\text{rank}(C_{\text{DMD}}) = m$.

(b) + (c) Since A_{DMD} has rank m and satisfies $A_{\text{DMD}}X = Y$ by the assumption [19], the image of A_{DMD} is given by

$$\text{im}(A_{\text{DMD}}) = \text{span}(x_1, \dots, x_m).$$

Therefore, the requirements of (b) and (c) in Theorem 1 are met for the matrix A_{DMD} .

(d) To show the equality $ZC_{\text{DMD}}Z^+ = A_{\text{DMD}}$, we first notice that both matrices solve the equation $AX = Y$ (see Theorem 1). Therefore, the two matrices are equal for vectors $x \in \text{span}(x_0, \dots, x_{m-1})$. In addition, this assertion is also true for the $m+1$ snapshot x_m because

$$\begin{aligned} ZC_{\text{DMD}}Z^+x_m &= ZC_{\text{DMD}}e_{m+1} = Z \begin{pmatrix} 0 \\ X^+x_m \end{pmatrix} \\ &= YX^+x_m = A_{\text{DMD}}x_m. \end{aligned}$$

Hence, $ZC_{\text{DMD}}Z^+x = A_{\text{DMD}}x$ for $x \in \text{span}(x_0, \dots, x_m)$. To conclude the proof, we show the equality for an arbitrary vector y from the orthogonal complement of $\text{span}(x_0, \dots, x_m)^\perp$:

$$\begin{aligned} (ZC_{\text{DMD}}Z^+)y &= ZC_{\text{DMD}}(Z^TZ)^{-1}Z^Ty \\ &= 0 \\ &= Y(X^TX)^{-1}X^Ty = YX^+y = A_{\text{DMD}}y. \end{aligned}$$

Since a subspace and its orthogonal complement span the entire space, the two matrices are equal. □

Theorem 1 and Theorem 2 show that a high-dimensional matrix A that solves the minimization problem 1 exactly, such as A_{DMD} , can be adequately represented by a low-dimensional representation capturing the data-relevant properties. Moreover, the low-dimensional representation has the structure of a companion matrix that naturally satisfies the first constraint in the minimization problem 6. By these characteristics, the high-dimensional constrained minimization problem 6 can be reformulated into the following low-dimensional constrained minimization problem (using the companion matrices from Theorem 1 and Theorem 2):

$$\begin{aligned} \min \quad & \|\tilde{c} - \tilde{c}_{\text{DMD}}\|_2^2 \\ \text{s.t.} \quad & \lambda_{\mathbb{I}} \text{ is eigenvalue of } \begin{bmatrix} 0^T & \tilde{c} \\ I_m & \tilde{c} \end{bmatrix}. \end{aligned} \quad (11)$$

(§3) The constraint in minimization problem 11 seems to have an impractical form as well. However, it can be solved in a feasible way: Since the characteristic polynomial of a companion matrix C with vector $\tilde{c}$ is given by

$$p_{\tilde{c}}(s) = -\tilde{c}_0 - \tilde{c}_1 s - \dots - \tilde{c}_m s^m + s^{m+1},$$

the constraint is equivalent to $p_{\tilde{c}}(1) = 0$ or, more precisely,

$$\tilde{c}_0 + \tilde{c}_1 + \dots + \tilde{c}_m = 1.$$

Using the notation $e_{m+1} = (1, 1, \dots, 1)^T \in \mathbb{R}^{m+1}$, the low-dimensional constrained minimization problem 11 is equivalent to

$$\begin{aligned} \min \quad & \|\tilde{c} - \tilde{c}_{\text{DMD}}\|_2^2 \\ \text{s.t.} \quad & e_{m+1}^T \tilde{c} = 1. \end{aligned} \quad (12)$$

The solution space $M_{\perp} = \{y \in \mathbb{R}^{m+1} ; e_{m+1}^T y = 1\}$ has several nice properties: It is an affine subspace and therefore convex, which implies existence and uniqueness of a solution for the constrained minimization problem 12 as the following theorem shows.

Theorem 3 *For the constrained minimization problem 12, there exists a unique solution given by the orthogonal projection $P_{\perp}$ onto $M_{\perp} = \{y \in \mathbb{R}^{m+1} ; e_{m+1}^T y = 1\}$. The projection can be computed via*

$$P_{\perp} y = y + \frac{1 - e_{m+1}^T y}{e_{m+1}^T e_{m+1}} e_{m+1}. \quad (13)$$

Proof Since the constraint is linear (more precisely, affine-linear), the solution space $M_{\perp}$ is an affine subspace. Hence, $M_{\perp}$ is a non-empty, closed, and convex set. This implies existence and uniqueness of a solution given by the orthogonal projection onto $M_{\perp}$. The explicit formula can be easily derived from the fact that $M_{\perp}$ is a hyperplane that is orthogonal to the vector e_{m+1} and passes through the point $r_0 = \frac{1}{m+1} \cdot e_{m+1} \in M_{\perp}$:

$$\begin{aligned} P_{\perp} y &= r_0 + \left(I_{m+1} - \frac{e_{m+1} e_{m+1}^T}{e_{m+1}^T e_{m+1}} \right) (y - r_0) \\ &= y + \frac{1 - e_{m+1}^T y}{e_{m+1}^T e_{m+1}} e_{m+1}. \end{aligned}$$

□

By Theorem 3, denoting $P_{\perp}$ as the (orthogonal) projection onto $M_{\perp}$, the solution to the constrained minimization problem 12 is given by

$$\tilde{c}_{\perp} = P_{\perp} \tilde{c}_{\text{DMD}} = \tilde{c}_{\text{DMD}} + \frac{1 - e_{m+1}^T \tilde{c}_{\text{DMD}}}{e_{m+1}^T e_{m+1}} \cdot e_{m+1}. \quad (14)$$

Algorithm 2 Efficient Exact Dynamic Mode Decomposition for Background Modeling (effexDMD4B)

```

1: function EFFEXDMD4B( $x_0, \dots, x_m, c_{\text{start}}, v_{\text{start}}$ )
2:    $X = [x_0 \dots x_{m-1}]$ 
3:    $c_{\text{DMD}} = \text{lsqr}(X, x_m, c_{\text{start}})$   $\triangleright \min \|Xc - x_m\|$ 
4:    $c_{\perp} = P_{\perp} \tilde{c}_{\text{DMD}} = P_{\perp} \begin{pmatrix} 0 \\ c_{\text{DMD}} \end{pmatrix}$ 
5:    $v_{\perp} = \text{power\_method}(C_{\perp}, v_0)$   $\triangleright C_{\perp} \dots C_{\perp} v_0$ 
6:    $\vartheta_{\perp} = [x_0 \dots x_{m-1} \ x_m] v_{\perp}$ 
7:   return  $\vartheta_{\perp}$ 
8: end function

```

The projection solves the robustness issues of DMD by enforcing the existence of an eigenvalue $\lambda_{\perp} = 1$ for the companion matrix $C_{\perp}$ to the vector $c_{\perp}$. The spectral properties of $C_{\perp}$, i.e., the eigenvector $v_{\perp}$ to the eigenvalue $\lambda_{\perp}$, can therefore be used to compute an accurate background model as illustrated in Figs. 2 and 3. The details will be explained in combination with the algorithm in the next subsection.

4.3 Algorithm

In the following, we describe our algorithm “efficient exact dynamic mode decomposition for background modeling” (effexDMD4B), shown in Algorithm 2. Although our algorithm is based on theorems of the previous subsection, it can be applied to any video data without any problems, even if the assumptions of the theorems do not hold.

The first step is to obtain the companion matrix C_{DMD} . By Eq. 10 (and the related Eq. 8), we have to compute $X^+ x_m$, which is the minimum-norm solution to the minimization problem

$$\min \|Xc - x_m\|_2^2. \quad (15)$$

This step is performed by an iterative least-squares method (lsqr), which has a computational complexity of $\mathcal{O}(nm)$. To keep the constant small, a special start value $c_{\text{start}} \in \mathbb{R}^{m+1}$ is used, adapted to the minimization problem 15 (Line 2–3). Due to the temporal coherence of video frames, the vector c_{start} should particularly emphasize the last data points. Moreover, it should be a partition of unity. We propose using

$$c_{\text{start}} = \left(\frac{1}{2^{m+1}}, \dots, \frac{1}{8}, \frac{1}{4}, \frac{1}{2} \right)^T \in \mathbb{R}^{m+1}, \quad (16)$$

which has shown fast convergence rates, even if the number of frames grow (as we will see later).

In Line 4 of Algorithm 2, the projection according to Eq. 14 is applied. This step enforces the existence of an eigenvalue $\lambda_{\perp}$.

Finally, the eigenvector $v_{\perp}$ to the eigenvalue $\lambda_{\perp}$ is computed by the power method (without normalization) and

Fig. 3 Our DMD approach for (grayscale) background modeling: After vectorizing the $m + 1$ video frames into n -dimensional data, the matrix C_{DMD} is computed and subsequently projected. Then, the power

method is applied to the low-dimensional and sparse companion matrix C_1 for the computation of background relevant DMD components that yield the background model

transformed to a mode (Line 5–6) that represents the background model statically. The power method uses the repeated application of the companion matrix C_1 (which is constructed via Eq. 8 and 9) to a start value v_{start} . This step allows extremely efficient computation at a complexity of $\mathcal{O}(m)$ because the companion matrix has a low-dimensional representation as well as a sparse structure with only $2m + 1$ non-zero entries.

Moreover, we propose using a special start value based on the temporal arithmetic mean $\bar{x} = \frac{1}{m+1} \sum_{k=0}^m x_k$ because the temporal mean represents a good initial guess for the background model. Since the power method operates in the low-dimensional space, the start vector is given by

$$v_{\text{start}} = Z^+ \bar{x} = \frac{1}{m+1} Z^+ \bar{x} = \frac{1}{m+1} e_{m+1}. \quad (17)$$

Although there is generally no guarantee that the power method converges for this start vector, our experiments have shown a stable convergence behavior. This is due to the fact that eigenvalues that do not belong to the background are typically located inside the unit circle (see Fig. 2).

4.4 Extensions

In the previous subsection, the basic algorithm was presented that is capable of computing grayscale background models that are sufficient for the most applications. However, our approach is versatile and can be extended to RGB data, video data with periodic parts, and to streaming data. Moreover, these extensions can be combined as well.

RGB Data Most background modeling algorithms only work with grayscale video frames because many formulations cannot handle multiple color channels at the same time. Moreover, the extensive computational overhead produced by additional data usually does not justify the slight improvements regarding accuracy. In contrast, our approach can be easily extended by keeping the computational overhead small. For this, we first follow Tirunagari et al. [35] to perform DMD on RGB-based video frames. Analogously to the left box of Fig. 3, the three channels of a RGB-based video

frame are vectorized and stacked on top of each other. Hence, the data for DMD has dimensionality $3n \times m + 1$. The remaining steps of our approach can be performed in the same way as shown in Fig. 3 or Algorithm 2 (note that the background model has the same organized structure as the RGB input data).

Data with periodic background parts Since DMD is based on a superposition principle (see Eq. 4 and 5), periodic parts in the background can be represented by additional DMD triples whose eigenvalues λ_j capture the frequency. As Fig. 4 shows, periodic processes are characterized by eigenvalues that lie on the unit circle. Hence, our approach can be easily extended in the following way:

Instead of computing only the static component, we use the power method for the ones that lie on the unit circle as well (see Fig. 4). More precisely, since the power method computes the greatest (in absolute value) eigenvalue, the shifted power method is used as long as values on the unit circle are detected (multiple applications with a fixed maximum still result in a computational complexity of $\mathcal{O}(m)$). Using the set

$$I_{1+} = \{j \in \{1, \dots, r_0\} ; |\lambda_j| - 1| < \delta\}$$

for a given threshold δ , the background of the video frames can be determined via (compare Eq. 5)

Fig. 4 Our extended DMD approach for background modeling. Using the power method multiple times, additional periodic parts in the background can be extracted. These periodic background components are characterized by DMD triples with eigenvalues on the unit circle (highlighted by orange circles)

$$x_k^{\text{BG}} = \sum_{j \in I_{\mathbb{1}+}} \lambda_j^k b_j \vartheta_j,$$

where the coefficients $b = (b_j)_{j \in I_{\mathbb{1}+}}$ need to be computed separately. To do this, we represent the background components by matrices, i.e.,

$$Z^{\text{BG}} = [x_0^{\text{BG}} \dots x_m^{\text{BG}}] = \Theta_{\mathbb{1}+} \text{diag}(b) V_{\mathbb{1}+},$$

where $\Theta_{\mathbb{1}+} = [\vartheta_j]_{j \in I_{\mathbb{1}+}}$ contains the respective modes column-wise and $V_{\mathbb{1}+}$ is the Vandermonde matrix to the eigenvalues $(\lambda_j)_{j \in I_{\mathbb{1}+}}$. Then, a minimization problem in the spirit of RPCA is used:

$$\min_b \|Z - \Theta_{\mathbb{1}+} \text{diag}(b) V_{\mathbb{1}+}\|_F.$$

The problem can be easily solved by the Moore-Penrose pseudoinverse as $I_{\mathbb{1}+}$ is typically very small:

$$\text{diag}(b) = \Theta_{\mathbb{1}+}^+ Z V_{\mathbb{1}+}^+.$$

Since the matrix is nearly diagonal, the off-diagonal elements have to be neglected. The resulting solution respects the complex conjugation of eigenvalues λ_j such that the scaled modes $b_j \vartheta_j$ occur in the same complex conjugate pairs. Consequently, the sum is real-valued and, therefore, we can readily generate a background model. This technique is illustrated with an example in Fig. 4. It can be observed that our extended approach captures the movement of the elevator precisely by additional two complex conjugated DMD components.

However, since the existence of additional appropriate eigenvalues (besides $\lambda_{\mathbb{1}} = 1$) is not guaranteed, this approach may select DMD components that do not belong to the background. Nevertheless, our experiments have shown that the procedure yields a robust algorithm if periodic patterns occur in the data (like the escalator in Fig. 4).

Streaming data for video surveillance Background modeling is often used for video surveillance, where data are continually acquired, i.e., it is assumed that $x_0, \dots, x_T$ is given (as well as a solution to that) and x_{T+1} is new. For the application of our approach to such streaming data, Algorithm 2 needs to be modified slightly. Whereas the projection (Line 4), the power method (Line 5), and the background model computation (Line 6) can be used in the same way, the least-squares method has to be altered.

In short, the problem is to find a solution $c_{\text{new}} \in \mathbb{R}^{T+1}$ for the minimization problem

$$\min \| [x_0 \dots x_T] c - x_{T+1} \|,$$

whereas the previous computed solution $c_{\text{old}} \in \mathbb{R}^T$ solves the minimization problem $\min \| [x_0 \dots x_{T-1}] c - x_T \|$. There

are two different techniques to tackle this problem, and both enable a streaming application in real time.

The first one uses an incremental approach for solving least-squares problems, e.g., an incremental singular value decomposition, for which many implementations are available. The second technique uses the least-squares method from Algorithm 2 with a modified start value that stems from the previous computation. We propose choosing

$$c_{\text{new}} = \frac{1}{2} \begin{pmatrix} c_{\text{old}} \\ 1 \end{pmatrix} \quad (18)$$

because c_{new} approximates the last relevant snapshot:

$$\begin{aligned} [x_0 \dots x_T] c_{\text{new}} &= \frac{1}{2} [x_0 \dots x_T] \begin{pmatrix} c_{\text{old}} \\ 1 \end{pmatrix} \\ &= \frac{1}{2} ([x_0 \dots x_{T-1}] c_{\text{old}} + x_T) \\ &\approx \frac{1}{2} x_T + \frac{1}{2} x_T \\ &= x_T. \end{aligned}$$

4.5 Implementation

Algorithm 2 was implemented in MATLAB R2019a, which already provides many optimized functions. For the least-squares method in Line 3, the internal function named “lsqr” was used with a start value according to Eq. 16. The maximum iterations were not limited. For the power method in the non-extended case, the companion matrix $C_{\mathbb{1}}$ was applied to the start value v_{start} (see Eq. 17) 100 times. Our tests have shown that this number of iteration produces accurate and robust background models. The procedure is done without normalization of the vector. Moreover, for the matrix vector multiplication the low-dimensional and sparse organized structure of the companion matrix should be taken into account (i.e., only $2m + 1$ entries are non-zero). For the power method in the extended case, the internal function “eigs” was used to compute various eigenvalues.

5 Experimental Evaluation

For the evaluation of our approach, two well-known datasets for foreground background separation were used: The CDnet 2014 dataset [38] and the I2R dataset [22] (used for the periodic background extension). We compare our approach effexDMD4B (both gray and RGB) with ten algorithms that stem from the LRSlibrary [32], which were chosen from different method-categories and speed-classes: ncR-CPA [18], R2PCP [17], FW-T [25], GROUSE [2], PG-RMC [8], RPCA-GD [40], pROST [15], OSTD [33], ROSL [31], and D-S-NMF [36]. The selection was also oriented toward

Fig. 5 Qualitative comparison of the results. From left to right: original frame (ORIG), groundtruth (GT), ncRPCA (a), R2PCP (b), FW-T (c), GROUSE (d), PG-RMC (e), RPCA-GD (f), pROST (g), OSTD (h),

ROSL (i), D-S-NMF (j), cDMD (k), exDMD (l), our algorithm gray effexDMD4B (m) and our algorithm RGB effexDMD4B (n). From top to bottom: see description of scenarios in Table 2

the results of the corresponding review by Bouwmans et al. [6]. Moreover, the traditional DMD approach (exDMD) [14] and the compressed variant (cDMD) [9] are evaluated as well.

The evaluation includes qualitative (see Fig. 5) and quantitative (see Table 1) experimental results as well as computation time measurements. For each algorithm, either the standard parameter setup was chosen (from the LRSlibrary) or the suggested settings from the associated publication. For the final foreground mask, a 5×5 median filter was used and the best result from four fixed thresholds were chosen to guarantee a fair comparison. The results were computed with MATLAB R2019a on a machine with 2.90GHz Intel Core i9-8950HK processor and 32GB RAM.

5.1 Accuracy

In Fig. 5, results from the above mentioned algorithms are compared qualitatively. An overview of the challenges of each scenario can be found in Table 2. Whereas some algorithms struggle with shadows (2nd row), ripping water (3rd row), or low framerates (9th and 10th row), our approach achieves adequate and robust results. For the bad weather (5th and 6th row) and thermal (7th and 8th row) scenarios, we achieve visually appealing foreground masks, in particular, for the 5th and 7th row. Only in the 1st and 4th row, our foreground mask (as well as the others) struggles with specific components of the frame, like glass, reflections, or spatial coherence.

These observations match to the quantitative outcomes in Table 1, where the F_1 -measure is computed for a selection of frames (where we have access to the ground truth). We observe that our approach both the grayscale version and the RGB version consistently obtain good results and remains robust, which is also true for GROUSE, RPCA-GD, D-S-NMF, and ROSL. In particular, the previous DMD approaches, cDMD and exDMD, often fail to detect foreground objects accurately (e.g., for canoe, skating, and tram), leading to scores less than 0.6. Moreover, many algorithms have problems in computing background models for bigger scenarios because computational intense matrix calculations have to be performed. To avoid these issues, either subsampling techniques need to be exploited (which we have not used) or partitioning techniques Table (2).

According to the quantitative analysis, our approach effexDMD4B can compete with other state-of-the-art algorithms in all scenarios. Whereas our RGB version achieves the best average scores along RPCA-GD, our grayscale approach has the fourth best average score after the algorithm ROSL. Other accurate algorithms are D-S-NMF, GROUSE, and pROST, achieving average scores higher than 0.8. The two other DMD-based background algorithms have the worst average scores due to their robustness issues. We therefore emphasize that enforcing an eigenvalue λ_1 resolves the robustness problems of DMD and leads to more accurate results. To substantiate this statement, several precision–recall curves are presented in Fig. 6. Whereas cDMD and exDMD often fail to score good precision and recall values,

Table 1 Quantitative comparison of the results in terms of the F_1 -measure

	LRSlibrary algorithms						DMD algorithms							
	ncRPCA [18]	R2PCP [17]	FW-T [25]	GROUSE [2]	PG-RMC [8]	RPCA-GD [40]	pROST [15]	OSTD [33]	ROSL [31]	D-S-NMF [36]	cDMD [9]	exDMD [14]	Our Gray	Our RGB
Highway	0.784	0.877	0.908	0.867	0.918	0.913	0.888	0.895	0.914	0.869	0.764	0.819	0.904	0.905
Pedestrians	0.850	0.743	0.778	0.807	0.855	0.855	0.852	0.633	0.855	0.850	0.814	0.703	0.852	0.862
Canoe	0.437	0.787	0.779	0.780	0.470	0.789	0.567	0.760	0.790	0.781	0.584	0.488	0.780	0.784
Fountain02	0.747	0.647	0.767	0.741	0.746	0.748	0.747	0.742	0.745	0.743	0.654	0.664	0.745	0.797
Blizzard	0.789	0.457	0.918	0.920	0.927	0.924	0.924	0.843	0.920	0.920	0.558	0.878	0.920	0.919
Skating	0.533	0.808	0.621	0.762	0.834	0.837	0.827	0.767	0.839	0.777	0.503	0.235	0.816	0.816
Corridor	0.568	0.681	0.719	0.800	0.786	0.858	0.827	0.787	0.857	0.804	0.782	0.854	0.851	0.851
Park	0.765	0.502	0.720	0.764	0.739	0.776	0.759	0.676	0.776	0.765	0.753	0.604	0.765	0.765
Tram	0.829	0.738	0.764	0.832	0.818	0.819	0.823	0.822	0.821	0.829	0.613	0.416	0.832	0.837
Turnpike	0.867	0.792	0.822	0.877	0.870	0.875	0.879	0.853	0.870	0.879	0.532	0.843	0.862	0.865
Average	0.717	0.703	0.780	0.815	0.796	0.840	0.809	0.778	0.839	0.822	0.656	0.650	0.833	0.840

The best scores are marked in boldface

Table 2 Description of chosen scenarios

Scenario	Size $\times$ No. frames	Challenges
Highway	$320 \times 240 \times 1550$	Fast moving cars
Pedestrians	$360 \times 240 \times 1099$	Slowly moving, shadows
Canoe	$320 \times 240 \times 1189$	Slowly moving, water
Fountain02	$432 \times 288 \times 1499$	Foreground fountain, water
Blizzard	$720 \times 480 \times 2001$	Fast moving cars, blizzard
Skating	$540 \times 360 \times 2900$	Skating people, heavy snowfall
Corridor	$320 \times 240 \times 3001$	Slowly moving people, thermal
Park	$352 \times 288 \times 600$	Slowly moving people, thermal
Tram	$640 \times 350 \times 900$	Tram, cars, low framerate: 1fps
Turnpike	$320 \times 240 \times 1500$	Cars, low framerate: 5fps

Fig. 6 Precision–recall curves for the three scenarios pedestrians, fountain02, and tram (from left to right), which are chosen from different categories. For comparison, the three most accurate algorithms RPCA-GD, ROSL, and D-S-NMF are selected as well as all other DMD-based methods **Fig. 7** Comparison of computation time (in seconds) for a test dataset with a resolution of 720×480 and 500 frames

we observe that our approach achieves accurate results for a variety of thresholds.

5.2 Computation Time

To demonstrate the efficiency of our algorithm, we first compare the overall computation time (except for the foreground mask procedure) of all algorithms for a given test dataset. The

test dataset is based on the blizzard scenario, has a resolution of 720×480 , and consists of 500 frames. The comparison is illustrated in Fig. 7. We observe that the grayscale version of effexDMD4B is extremely efficient calculating the background model in nearly 6 seconds. The RGB version needs approximately three times as much computation time. This highlights the fact that Algorithm 2 evolves linearly in space because effexDMD4B_RGB has to deal with three color channels. The traditional DMD algorithm exDMD requires nearly 35 times more computation time than our algorithm (gray), whereas cDMD only needs 1.5 times more computation time due to its high compression rate. For the algorithms from the LRSlibrary, D-S-NMF is almost as fast as our approach. The other algorithms, like ncRPCA, FW-T, and ROSL (which also belong to the highest speed classification), are at least an order of magnitude slower.

Besides the absolute computation time, we also want to demonstrate how the algorithm scales when the number of frames grows. Therefore, Fig. 8 shows the computation time for the six most efficient algorithms as a function of number of frames. We observe that our approach effexDMD4B

Fig. 8 Comparison of computation time (in seconds) as function of number of frames for the five most efficient algorithms

(grayscale and RGB) as well as Deep-Semi-NMF evolve linearly as the lines coincide with their dashed lines (linear least-squares fit). Previous DMD-based methods, like cDMD (or exDMD), highly suffer from operations that exhibit quadratic and cubic computational complexity (due to the computation of all DMD components while our approach avoids it). This can be observed by the purple dashed line, which does not match with the solid purple line. A similar behavior can be observed for ROSL and even worse for FW-T.

6 Discussion and Conclusion

In this paper, we have presented a DMD-based approach for the computation of background models. According to the qualitative and quantitative results, our approach can compete with other state-of-the-art methods. It produces accurate background models (just like RPCA-GD and ROSL) and remains robust, which is a significant improvement compared to previous DMD-based algorithms.

However, the real strengths of effexDMD4B are its high efficiency and flexibility. It is extremely fast and additionally evolves linearly in time and space, whereas most other algorithms exhibit quadratic or even cubic computational complexities. Moreover, it can be applied to a wide range of scenarios with different challenges without adjustment. Many background modeling algorithms need to modify parameter settings to produce accurate results. Another sort of flexibility is the extension of our approach to RGB data, data with periodic backgrounds parts, and to streaming data.

Since DMD offers great potential, we plan to extend our approach to video data that exhibit panning, zooming, jittering, and intermitting elements. This may include the use of different pre- and post-processing tools, like spatio-temporal smoothing, to obtain a more accurate foreground mask. Another idea is to apply other iterative least-squares methods (lsqr) that respect the special temporal coherence structure of our minimization problem. This consideration is

particularly important for the application to streaming data, where special emphasis is on the least-squares methods (since an execution is needed in every update step). Moreover, since DMD can deal with compressed data (due to the spectral decomposition), a version of effexDMD4B with compression techniques may further boost efficiency.

Acknowledgements This work is partly supported by “Kooperatives Promotionskolleg Digital Media” at Hochschule der Medien and the University of Stuttgart. Besides that, it is funded by the “Deutsche Forschungsgemeinschaft” (DFG, German Research Foundation) - Project-ID 251654672 - TRR 161 (B01, B04).

Funding Open Access funding enabled and organized by Projekt DEAL.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Akilan, T., Wu, Q.J., Jiang, W., Safaei, A., Huo, J.: New trend in video foreground detection using deep learning. In: IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS), pp. 889–892 (2018). <https://doi.org/10.1109/MWSCAS.2018.8623825>
2. Balzano, L., Nowak, R., Recht, B.: Online identification and tracking of subspaces from highly incomplete information. In: 48th Annual Allerton Conference on Communication, Control, and Computing, pp. 704–711 (2010). <https://doi.org/10.1109/ALLERTON.2010.5706976>
3. Bi, C., Yuan, Y., Zhang, J., Shi, Y., Xiang, Y., Wang, Y., Zhang, R.: Dynamic mode decomposition based video shot detection. IEEE Access **6**, 21397–21407 (2018). <https://doi.org/10.1109/ACCESS.2018.2825106>
4. Bouwmans, T., Javed, S., Sultana, M., Jung, S.K.: Deep neural network concepts for background subtraction: A systematic review and comparative evaluation. Neural Netw. **117**, 8–66 (2019). <https://doi.org/10.1016/j.neunet.2019.04.024>
5. Bouwmans, T., Porikli, F., Höferlin, B., Vacavant, A.: Background modeling and foreground detection for video surveillance. CRC Press, USA (2014)
6. Bouwmans, T., Sobral, A., Javed, S., Jung, S.K., Zahzah, E.H.: Decomposition into low-rank plus additive matrices for background/foreground separation. Computer Sci. Rev. **23**, 1–71 (2017). <https://doi.org/10.1016/j.cosrev.2016.11.001>
7. Braham, M., Van Droogenbroeck, M.: Deep background subtraction with scene-specific convolutional neural networks. In: 2016 International Conference on Systems, Signals and Image Processing (IWSSIP), pp. 1–4 (2016). 10.1109/IWSSIP.2016.7502717

8. Cherapanamjeri, Y., Gupta, K., Jain, P.: Nearly optimal robust matrix completion. In: International Conference on Machine Learning (ICML), pp. 797–805 (2017). <https://doi.org/10.5555/3305381.3305464>
9. Erichson, N.B., Brunton, S.L., Kutz, J.N.: Compressed dynamic mode decomposition for background modeling. *J. Real-Time Image Process. (JRTIP)* **16**(5), 1479–1492 (2019). <https://doi.org/10.1007/s11554-016-0655-2>
10. Erichson, N.B., Donovan, C.: Randomized low-rank dynamic mode decomposition for motion detection. *Computer Vision Image Understand. (CVIU)* **146**, 40–50 (2016). <https://doi.org/10.1016/j.cviu.2016.02.005>
11. Garcia-Garcia, B., Bouwmans, T., Silva, A.J.R.: Background subtraction in real applications: challenges, current models and future directions. *Computer Sci. Rev.* (2020). <https://doi.org/10.1016/j.cosrev.2019.100204>
12. Giraldo, J.H., Bouwmans, T.: Graphbgs: Background subtraction via recovery of graph signals. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 6881–6888 (2021). doi: <https://doi.org/10.1109/ICPR48806.2021.9412999>
13. Giraldo, J.H., Javed, S., Bouwmans, T.: Graph moving object segmentation. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **56**, 1–10 (2020). <https://doi.org/10.1109/TPAMI.2020.3042093>
14. Grosek, J., Kutz, J.N.: Dynamic mode decomposition for real-time background/foreground separation in video. <http://arxiv.org/abs/1404.7592> (2014)
15. Hage, C., Kleinstaub, M.: Robust PCA and subspace tracking from incomplete observations using ℓ_0 -surrogates. *Comput. Stat. (CompStat)* **29**(3–4), 467–487 (2014). <https://doi.org/10.1007/s00180-013-0435-4>
16. Haq, I.U., Fujii, K., Kawahara, Y.: Dynamic mode decomposition via dictionary learning for foreground modeling in videos. *Computer Vision Image Understand. (CVIU)* (2020). <https://doi.org/10.1016/j.cviu.2020.103022>
17. Hintermüller, M., Wu, T.: Robust principal component pursuit via inexact alternating minimization on matrix manifolds. *J. Math. Imag. Vision (JMIV)* **51**(3), 361–377 (2015). <https://doi.org/10.1007/s10851-014-0527-y>
18. Kang, Z., Peng, C., Cheng, Q.: Robust pca via non-convex rank approximation. In: IEEE International Conference on Data Mining (ICDM), pp. 211–220 (2015). <https://doi.org/10.1109/ICDM.2015.15>
19. Krake, T., Weiskopf, D., Eberhardt, B.: Dynamic mode decomposition: Theory and data reconstruction. <http://arxiv.org/abs/1909.10466> (2019)
20. Kutz, J.N., Brunton, S.L., Brunton, B.W., Proctor, J.L.: Dynamic mode decomposition data-driven modeling of complex systems. *SIAM Soc. Indus. Appl. Math.* **2**, 51 (2016). <https://doi.org/10.1137/1.9781611974508>
21. Kutz, J.N., Fu, X., Brunton, S.L., Erichson, N.B.: Multi-resolution dynamic mode decomposition for foreground/background separation and object tracking. *IEEE International Conference on Computer Vision Workshop (ICCVW)* pp. 921–929 (2015). <https://doi.org/10.1109/ICCVW.2015.122>
22. Li, L., Huang, W., Gu, I.Y.H., Tian, Q.: Statistical modeling of complex backgrounds for foreground object detection. *IEEE Trans. Image Process. (TIP)* **13**(11), 1459–1472 (2004). <https://doi.org/10.1109/TIP.2004.836169>
23. Mandal, M., Vipparthi, S.K.: An empirical review of deep learning frameworks for change detection: Model design, experimental frameworks, challenges and research needs. *IEEE Transactions on Intelligent Transportation Systems* (2021). (to appear)
24. Minematsu, T., Shimada, A., Taniguchi, R.I.: Rethinking background and foreground in deep neural network-based background subtraction. In: 2020 IEEE International Conference on Image Processing (ICIP), pp. 3229–3233 (2020). <https://doi.org/10.1109/ICIP40778.2020.9191151>
25. Mu, C., Zhang, Y., Wright, J., Goldfarb, D.: Scalable robust matrix recovery: Frank-wolfe meets proximal methods. *SIAM J. Scientific Comput. (SISC)* **38**(5), A3291–A3317 (2016). <https://doi.org/10.1137/15M101628X>
26. Narayanamurthy, P., Vaswani, N.: A fast and memory-efficient algorithm for robust pca (merop). In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4684–4688 (2018). <https://doi.org/10.1109/ICASSP.2018.8461540>
27. Ngo, T.T., Nguyen, V., Pham, X.Q., Hossain, M.A., Huh, E.N.: Motion saliency detection for surveillance systems using streaming dynamic mode decomposition. *Symmetry* **12**(9), 1397 (2020). <https://doi.org/10.3390/sym12091397>
28. Pendergrass, S.D., Brunton, S.L., Kutz, J.N., Erichson, N.B., Askham, T.M.: Dynamic mode decomposition for background modeling. In: IEEE International Conference on Computer Vision Workshops (ICCVW), pp. 1862–1870 (2017). <https://doi.org/10.1109/ICCVW.2017.220>
29. Rodriguez, P., Wohlberg, B.: Incremental principal component pursuit for video background modeling. *J. Math. Imag. Vision (JMIV)* **55**(1), 1–18 (2016). <https://doi.org/10.1007/s10851-015-0610-z>
30. Schmid, P., Sesterhenn, J.: Dynamic mode decomposition of numerical and experimental data. In 61st Annual Meeting of the APS Division of Fluid Dynamics. American Physical Society **53**(15) (2008)
31. Shu, X., Porikli, F., Ahuja, N.: Robust orthonormal subspace learning: Efficient recovery of corrupted low-rank matrices. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3874–3881 (2014). <https://doi.org/10.1109/CVPR.2014.495>
32. Sobral, A., Bouwmans, T., hadi Zahzah, E.: LRSLibrary: Low-rank and sparse tools for background modeling and subtraction in videos. *Robust Low-Rank and Sparse Matrix Decomposition: Applications in Image and Video Processing* (2016)
33. Sobral, A., Javed, S., Jung, S.K., Bouwmans, T., hadi Zahzah, E.: Online stochastic tensor decomposition for background subtraction in multispectral video sequences. In: Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW), pp. 106–113 (2015). <https://doi.org/10.1109/ICCVW.2015.125>
34. Sobral, A., Vacavant, A.: A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. *Computer Vision Image Understand. (CVIU)* **122**, 4–21 (2014). <https://doi.org/10.1016/j.cviu.2013.12.005>
35. Tirunagari, S., Poh, N., Bober, M., Windridge, D.: Can dmd obtain a scene background in color? In: International Conference on Image, Vision and Computing (ICIVC), pp. 46–50 (2016). <https://doi.org/10.1109/ICIVC.2016.7571272>
36. Trigeorgis, G., Bousmalis, K., Zafeiriou, S., Schuller, B.W.: A deep matrix factorization method for learning attribute representations. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **39**(3), 417–429 (2016). <https://doi.org/10.1109/TPAMI.2016.2554555>
37. Vaswani, N., Bouwmans, T., Javed, S., Narayanamurthy, P.: Robust subspace learning: robust pca, robust subspace tracking, and robust subspace recovery. *IEEE Signal Process. Magaz.* **35**(4), 32–55 (2018). <https://doi.org/10.1109/MSP.2018.2826566>
38. Wang, Y., Jodoin, P.M., Porikli, F., Konrad, J., Benzeeth, Y., Ishwar, P.: Cdn2014: An expanded change detection benchmark dataset. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 393–400 (2014). <https://doi.org/10.1109/CVPRW.2014.126>
39. Xu, Y., Dong, J., Zhang, B., Xu, D.: Background modeling methods in video analysis: a review and comparative evaluation. *CAAI Trans. Intell. Technol.* **1**(1), 43–60 (2016). <https://doi.org/10.1016/j.trit.2016.03.005>

40. Yi, X., Park, D., Chen, Y., Caramanis, C.: Fast algorithms for robust PCA via gradient descent. *Adv. Neural Inf. Process Syst. (NIPS)* **29**, 4152–4160 (2016). <https://doi.org/10.5555/3157382.3157562>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Tim Krake is a PhD student at the Visualization Research Center, University of Stuttgart and at the Hochschule der Medien, Stuttgart, where he is a member of the joint graduate school Digital Media. He is currently working on the integration of mathematical techniques (mainly based on spectral decomposition methods) to the field of computer graphics and visualization. In this context, his particular interest is on Dynamic Mode Decomposition (DMD). He received both MSC

and BSc in mathematics at the University of Tuebingen.

Andrés Bruhn received his Ph.D. in Computer Science from Saarland University in Germany in 2006. After working as a postdoctoral researcher from 2006–2007 he was associated as an assistant professor with the Mathematical Image Analysis Group at Saarland University (2007–2012). During this time he built up an independent research group on Vision and Image Processing within the Cluster of Excellence “Multimodal Computing and Interaction” (2010–2012). Since 2012 he holds the

position of a full professor of Computer Science at University of Stuttgart, where he heads the Computer Vision group at the Institute for Visualization and Interactive Systems. His research interests include various areas in the field of computer vision, image processing, machine learning and optimization. In this context, he is particularly interested in motion estimation and geometric computer vision. His publications received several award such as an ECCV best paper award, the DAGM Olympus prize, the GI/SI/OCG/GChACM Outstanding Dissertation award, an ERCIM Cor Baayen Award nomination as well as the ECCV Koenderink Prize for Fundamental Contributions in Computer Vision.

Bernhard Eberhardt received a masters degree in mathematics from the University of Massachusetts at Amherst and a Diploma in mathematics from the University of Tuebingen. In 1994 he received a PhD in mathematics from the University of Tuebingen and completed his habilitation in computer graphics in 2001 under the guidance of Dr. Wolfgang Strasser. Currently, Dr. Eberhardt is full professor at Hochschule der Medien, Germany and head of the computer animation

department. His research interests include computer animation, visualization, computer graphics and image processing as well as computer vision. His main research areas are in physically based modeling, motion capture and image based rendering. Since 2012 he is leading the cooperative doctoral program Digital Media between the Universities of Tübingen, Stuttgart and Hochschule der Medien. He is a member of Eurographics, Gesellschaft für Informatik and ACM-SIGGRAPH.

Daniel Weiskopf is a Professor at the Visualization Research Center (VISUS) of the University of Stuttgart, Germany. He received his Dr.rer.nat. degree (similar to PhD) in Physics from the University of Tübingen, Germany, in 2001, and the Habilitation degree in Computer Science from the University of Stuttgart in 2005. His research interests include visualization, visual analytics, eye tracking, human-computer interaction, computer graphics, and special and general relativity.