AIAA SciTech Forum
6-10 January 2020, Orlando, FL
AIAA Scitech 2020 Forum

10.2514/6.2020-1943

Multi-Sensor Scene Segmentation for Unmanned Air and Ground Vehicles using Dynamic Mode Decomposition

Aniketh Kalur*, Debraj Bhattacharjee†, Youbing Wang‡ and Maziar S. Hemati§
University of Minnesota, Minneapolis, MN 55455, USA

Vibhor L. Bageshwar¶
Honeywell Aerospace, Honeywell International, Plymouth MN, 55441, USA

Situational awareness is the key to safe navigation and operation in autonomous air and ground systems. Multi-sensor scene segmentation can facilitate situational awareness for autonomous navigation. In this work, we introduce an alternative to learning-based approaches for scene segmentation based on dynamical systems theory. The streaming dynamic mode decomposition (DMD) algorithm is tailored to isolate moving foreground objects from stationary background objects in the scene. We show that LiDAR point cloud data can be transformed into a cylindrical depth map that is amenable to analysis within the DMD framework. We further extend the streaming DMD algorithm to adapt to changes in the scene by introducing a forgetting factor that weights the relative importance of past versus present information in the data stream. The proposed streaming DMD methods are applied for on-the-fly scene segmentation of video and LiDAR data streams from the KITTI Dataset. We find that the approach is able to distinguish between stationary background and moving foreground objects in a consistent and reliable manner across various scenes.

I. Introduction

Autonomous systems operating in urban environments face several challenges in detecting and identifying surrounding objects. Situational awareness is paramount for autonomous systems, especially when operating in tandem with humans. A large number of sensors, such as vision, infrared, and sound navigation and ranging (SONAR), have been used to help improve situational awareness in cluttered and dynamically evolving environments. Gaining situational awareness from rich data streams from multiple on-board sensors is of great importance. The advent of low-cost light detection and ranging (LiDAR) sensors has enabled 360° scene reconstruction with precise distance measurements. LiDAR facilitates simultaneous localization and mapping (SLAM), object detection, and range measurement. Due to their small size and precise ranging capabilities, LiDAR sensors are now ubiquitous in a variety of autonomous systems, including unmanned aerial and ground vehicles.

Despite its advantages, LiDAR introduces difficulties when used for object detection, isolation, and identification. In contrast to information from vision systems, where the image obtained is the pixel intensity for every spatial location in the field of view (FOV) on a fixed grid, LiDAR sensors predominantly obtain range and intensity of reflections from every point on which the light impinges. Despite obtaining high resolution and precise range information, working with LiDAR sensors poses an inherent challenge: These sensors generate large amounts of data from each scan—the number of points is generally $\sim O(10^5)$. These large point clouds are sensitive to aspects like the angle of incidence and reflection of beams, distance from the scanner, and lack of texture information. Since processing and interpreting large 3D point cloud data can be challenging, many processing strategies rely upon a projection of the 3D point cloud onto a 2D plane. This facilitates processing LiDAR information by leveraging existing and well-established techniques from the vision systems community [1, 2]. Apart from size and robustness challenges, LiDAR point cloud data also present processing challenges for object registration and moving object detection.

An essential precursor to many of the feature recognition, object tracking, and object classification methods is the separation of a dynamic scene into stationary background and moving foreground components. Isolating background

*Graduate Student, Aerospace Engineering and Mechanics, University of Minnesota. AIAA Student Member.
†Graduate Student, Electrical and Computer Engineering, University of Minnesota. AIAA Student Member.
‡Research Scientist, College of Science and Engineering, University of Minnesota.
§Assistant Professor, Aerospace Engineering and Mechanics, University of Minnesota. AIAA Senior Member.
¶Research Scientist, Advanced Technology, Honeywell Aerospace, Honeywell International. AIAA Senior Member.

and foreground information from video and LiDAR point cloud data can facilitate these higher-level tasks by isolating dynamically relevant features for further processing and interpretation. Numerous approaches have been proposed for background/foreground separation, with robust principal component analysis (RPCA) being a state-of-the-art method in video processing applications [3]. Recently, Grosek et al. proposed a data-driven dynamical systems approach for dynamic scene segmentation that performs competitively with RPCA in video applications [4]. The approach leverages a sliding window formulation of the dynamic mode decomposition (DMD) algorithm to distill time-sequences of snapshot data—video frames in this case—into dynamically relevant spatio-temporal modes in the scene [5, 6]. In doing so, moving foreground and stationary background can be distinguished from one another unambiguously.

Motivated by the demonstrated utility of DMD-based approaches for scene segmentation in video processing applications, the present study aims to extend such capabilities to the multi-sensor setting. LiDAR point cloud data, in particular, is fundamentally different in nature to pixel-based RGB data. Raw LiDAR point cloud data are not suitable for scene segmentation within the DMD framework. To overcome this issue, we propose transforming the 3D point cloud data into a 2D depth map over a cylindrical surface with a 360° field of view (FOV). The 2D depth map is analogous to grayscale image data from a vision system, thereby enabling the dynamic scene segmentation of LiDAR data using DMD-based methods. Since real-time processing is an important consideration for operational autonomy, we propose an adaptive streaming DMD algorithm to facilitate "on-line" scene segmentation. The proposed approach extends the streaming DMD algorithm proposed in [7] to better adapt to changes in the environment, as needed in autonomous navigation applications. The streaming DMD approach is an excellent fit for scene segmentation applications in autonomous systems, as it enables the extraction of dynamical scene information from incoming data streams in real-time. The streaming DMD perspective also provides a flexible alternative to the sliding window DMD approach proposed in [4]. Finally, we show that DMD-based algorithms for scene segmentation in the multi-sensor setting of video and LiDAR provide complementary perspectives of the same segmented scene, which can be used to inform higher-level tasks. The proposed algorithms are analyzed and demonstrated on vision and LiDAR data from the KITTI Dataset [8].

The paper is organized as follows: In Section II, we provide background on dynamic scene segmentation and discuss previous works that have leveraged DMD for such tasks in video processing applications. Preparation of both video and LiDAR data for DMD-based scene segmentation is detailed in Section III. In Section IV.A, we introduce the original streaming DMD algorithm and a more versatile adaptive streaming DMD algorithm. We investigate and discuss the scene segmentation performance of each streaming DMD method in Section V. Finally, conclusions are provided in Section VI.

II. Background and previous work

Object detection and classification can benefit from dynamic scene segmentation algorithms, in which objects with distinct dynamics are differentiated from one another. Generally, dynamic models are designed using learning-based approaches; however, learning the dynamics of objects in a scene can be very challenging. To this end, many studies have used unsupervised learning approaches. For example, exemplar-based unsupervised learning has been proposed for classifying dynamic objects [9]. Further, much of the work on supervised and unsupervised learning has been performed on vision-based systems [10, 11]. Although effective techniques have been developed for learning information from LiDAR sensors [2, 12–14], the resources required for implementing such approaches can be unwieldy. For video applications, existing classifiers can be leveraged and extended for a specific application with relative ease. In contrast, obtaining data and training classifiers for LiDAR-based machine learning approaches currently requires a significant upfront cost in terms of time and effort.

An attractive alternative to these learning-based approaches is to leverage data-driven dynamical systems theory to extract dynamically relevant information in the scene on-the-fly. Indeed, scene information from registered LiDAR and vision sensors can be interpreted as being generated by a dynamical system. This dynamical systems interpretation of a scene is advantageous because it allows objects in the scene to be differentiated according to their dynamics, without the upfront costs associated with training that are requisite of learning-based approaches. The dynamic mode decomposition (DMD) is one such approach whose utility in background and foreground separation has been demonstrated within the context of video data [4]. DMD is a data-driven "batch-processing" technique that has been widely used in the fluid dynamics community to extract spatio-temporal modes from complex and dynamically evolving datasets [5, 6]. In their study of DMD-based video scene segmentation, Grosek et al. partition the video data into smaller batches before performing DMD. This results in an on-line "sliding window" approach that can be used to distill dynamical information from video streams [4]. This sliding-window DMD approach is proposed as a competitive

alternative to other state-of-the-art background and foreground separation methods. For example, the DMD approach compares favorably with robust principal component analysis (RPCA)—a convex optimization approach [3]—for vision-based scene segmentation.

DMD operates on empirical snapshot data to extract rich dynamical information that can be used to segment a scene according to its underlying dynamics. For video applications, the snapshots would simply be RGB data from individual frames of the video feed. Denoting the snapshot at time t_k as $\mathbf{x}(t_k) \in \mathbb{R}^n$, DMD approximates the eigenvectors and eigenvalues of a linear dynamical system A that maps $\mathbf{x}(t_k)$ to $\mathbf{x}(t_{k+1})$ (i.e., A advances one frame of the video to the next in the sequence). The DMD eigenvectors provide information about the spatial character of a mode, while the DMD eigenvalues provide information about the temporal behavior of that same mode. Since DMD assumes a linear dynamical system, each mode exhibits simple temporal dynamics that can be characterized by a single frequency and a single growth/decay rate [15, 16]. It was shown in [6] that the linear map A is simply the minimum-norm/least-squares solution to $AX = Y$, where $X := \begin{bmatrix}\mathbf{x}(t_0) & \mathbf{x}(t_2) & \dots & \mathbf{x}(t_{m-1})\end{bmatrix} \in \mathbb{R}^{n\times m}$ and $Y := \begin{bmatrix}\mathbf{x}(t_1) & \mathbf{x}(t_3) & \dots & \mathbf{x}(t_m)\end{bmatrix} \in \mathbb{R}^{n\times m}$. DMD determines the eigenvalues and eigenvectors of this linear map in an efficient manner by leveraging a reduced singular value decomposition (SVD) of X. Specifically, if $X = U\Sigma V^*$, then a lower-dimensional proxy system $\tilde{A} = U^*AU$ can be formed, where U contains the leading r left singular vectors of X and $\tilde{A} \in \mathbb{R}^{r\times r}$. Then, the eigendecomposition of this low-dimensional proxy system $\tilde{A}\mathbf{w}_k = \mathbf{w}_k\lambda_k$ can be related to the eigenvectors $\phi_k = U\mathbf{w}_k$ and non-zero eigenvalues λ_k of A. Further details of the DMD algorithm and interpretation of its outputs can be found in [6, 15, 16].

Since DMD eigenvalues reveal temporal information in the makeup of a scene, the information regarding the background and foreground can be found by examining the frequency of each DMD mode $\omega_k = \angle\lambda_k/2\pi\Delta t$, where Δt is the sampling interval between snapshots. As suggested in [4], the DMD modes with $\omega_k \approx 0$ correspond to slowly time-varying—approximately stationary—background information that makes up the "low-rank" component L in the scene. To obtain the low-rank component, we select $|\omega_k| \leq \epsilon \ll 1$, where ϵ is some threshold value that is application-specific. Similarly, the foreground information comprises a "sparse" component S in the scene. This sparse foreground component can be obtained by selecting the remaining ω_k values. The initial amplitude b_k of each mode is given by the inner-product between the first snapshot $\mathbf{x}(t_0)$ and the k^{th} reciprocal DMD mode (i.e., left eigenvector of A associated with λ_k). The low-rank (L) and sparse (S) components of a scene obtained from DMD are given by [4],

$$\begin{aligned} L &\approx \sum_{|\omega_k|\leq\epsilon} b_k\phi_k e^{\omega_k t} \\ S &\approx \sum_{|\omega_k|>\epsilon} b_k\phi_k e^{\omega_k t}. \end{aligned} \tag{1}$$

In this study, we extend the background-foreground separation method proposed in [4] to the context of LiDAR data. A key step in extending the approach for scene segmentation with LiDAR is in preparing the LiDAR point cloud data to a format that is appropriate for the analysis within the DMD framework. Details about data preparation are presented in the next section. In subsequent sections, we show that a streaming formulation of the DMD algorithm can be used to efficiently perform video and LiDAR scene segmentation on-the-fly, as is needed for real-world operation.

III. Data preparation

In this work, we use grayscale video and LiDAR point cloud data from the "city" and "campus" categories of the KITTI Dataset [8]. Video data preparation for DMD is straightforward. Video data is a set of time sequenced images captured at equally spaced time intervals, so only snapshot matrices X and Y need be formed for use within the DMD framework. First, each grayscale image in the sequence is vectorized $\mathbf{x}(t_k)$ and stored as a column of the snapshot matrix X (see Figure1). The corresponding time-shifted snapshots $\mathbf{x}(t_{k+1})$ are stored as columns of the shifted snapshot matrix Y. Then, the DMD algorithm can be applied directly to the snapshot data matrices in X and Y to extract spatio-temporal modes.

In contrast to the 2D spatially fixed grayscale images provided by a camera (i.e., an Eulerian frame), a single LiDAR scan provides 3D point cloud data in a spatially varying frame (i.e., a Lagrangian frame). For every beam impinging on an object, the LiDAR sensor provides range information of the object relative to itself, which is then converted to Cartesian coordinates relative to the scanner. The LiDAR sensor provides accurate ranging information with a 360° FOV; whereas traditional vision sensors have a restricted field of view (see Figure 2a). However, due to spatial variance of the point cloud, LiDAR data cannot be directly processed within the DMD framework.

To enable DMD-based foreground-background separation of the LiDAR information, we convert the 3D point cloud

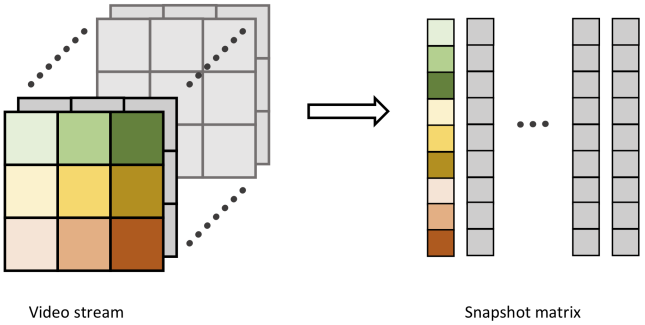


Figure 1 Vectorizing video data for streaming DMD

data to a flattened spatially fixed format, akin to data from a camera. To do so, the point cloud data is first projected onto a cylindrical surface. Then, the cylinder is unwrapped to obtain a 2D depth map (see Figure 3). The resulting depth map is similar to an image: Instead of pixel intensity, the 2D depth map contains depth information relative to LiDAR sensor (see Figure 2b).

The projection of individual points $P(x_i, y_i, z_i)$ is obtained as,

$$\theta_i = atan2(y_i, x_i) \tag{2}$$

$$\phi_i = arcsin(z_i/\sqrt{x_i^2 + y_i^2 + z_i^2}) \tag{3}$$

where (x_i, y_i, z_i) denotes the Cartesian coordinates of the i^{th} point in 3D space. Once the points are projected onto a cylindrical surface, we create a 2D map to store the corresponding depth information. The 2D depth map information associated with the point of interest $P(x_i, y_i, z_i)$ is stored using an alternative indexing (r_i, c_i) on the cylinder surface, where $r_i := [\phi_i/\Delta\phi]$ and $c_i := [\theta_i/\Delta\theta]$. Here, θ is the azimuth angle and ϕ is the elevation angle. Further, $\Delta\theta$ and $\Delta\phi$ denote the resolution of the azimuth and elevation angle, respectively. In this 2D map, we fill the index (r_i, c_i) with the associated depth $d_i = \sqrt{x_i^2 + y_i^2}$ associated with $P(x_i, y_i, z_i)$. An illustration of the depth map obtained from a 3D LiDAR point cloud is shown in Figure 2b.

Once a depth map is created for every LiDAR scan, we obtain a set of time-sequenced depth maps. Each of these depth maps is then vectorized, as done with images. We then form the time-shifted snapshot matrices (X, Y) of these depth maps. Now that the LiDAR data has been processed in a format that can be handled within the DMD framework, the process of application of DMD-based scene segmentation algorithms can proceed similarly to the case of video data.

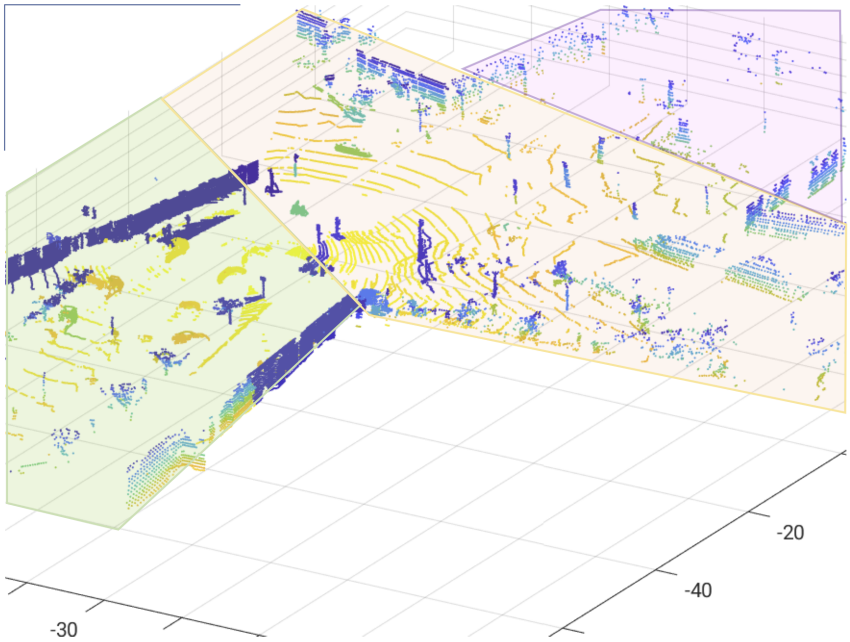
IV. Streaming DMD for background-foreground separation in video and LiDAR

In this work, we use a streaming DMD algorithm for background and foreground separation from video and registered LiDAR data streams. The streaming variant of DMD extends itself directly to autonomous systems; as autonomous systems often need to operate from sensory data available incrementally in streams. The streaming DMD method is capable of updating DMD computations incrementally (as data streams become available) while retaining performance similar to the 'batch-processed' DMD [7]. This section describes the streaming DMD formulation and adaptive streaming DMD formulation.

A. Streaming DMD Algorithm

The streaming DMD method helps process information incrementally when a new snapshot becomes available. This ability is a paradigm shift for processing data obtained from sensors on-board an autonomous vehicle. Sensors on-board autonomous systems scan the environment in sequence, where data from the current scan will have correlation to data obtained from the previous scan. The ability to process data incrementally is a tremendous advantage; specifically when the accuracy of the extracted information is mathematically equivalent to the batch-processing method like the DMD. Therefore, streaming DMD algorithm is practical in applications where data is being streamed online in 'real-time'.

To update the DMD operator with new incoming snapshots, we use a streaming variant of the DMD algorithm that updates the linear operator when incoming snapshots becoming available. Consider a single snapshot pair $(\mathbf{x}_i, \mathbf{y}_i) =$



(a) LiDAR point cloud

(b) Projected LiDAR depth map

(c) video

Figure 2 3D LiDAR point cloud (Figure 2a), here objects are clustered by Euclidean distance for better viewing. The corresponding cylindrical depth map and camera field of view are shown in Figure 2b and Figure 2c respectively. Specific areas in the point cloud, depth map and image from camera have been highlighted with a color code. The colored highlights (green, yellow, violet) in upper panel correspond to three sectors, the corresponding three sectors locations have been shown in the depth map. The image in Figure 2c correspond to the scene in center pane of Figure 2b

$(\mathbf{x}(t_k), \mathbf{x}(t_{k+1}))$ corresponding to columns of X and Y, respectively. Then, from Section II, the high-dimensional DMD operator $A \in \mathbb{R}^{n\times n}$ can be projected into a low-dimensional proxy system using an orthonormal basis for the image of X (denoted Q_x), i.e.,

$$\tilde{A} = Q_X^T A Q_X. \tag{4}$$

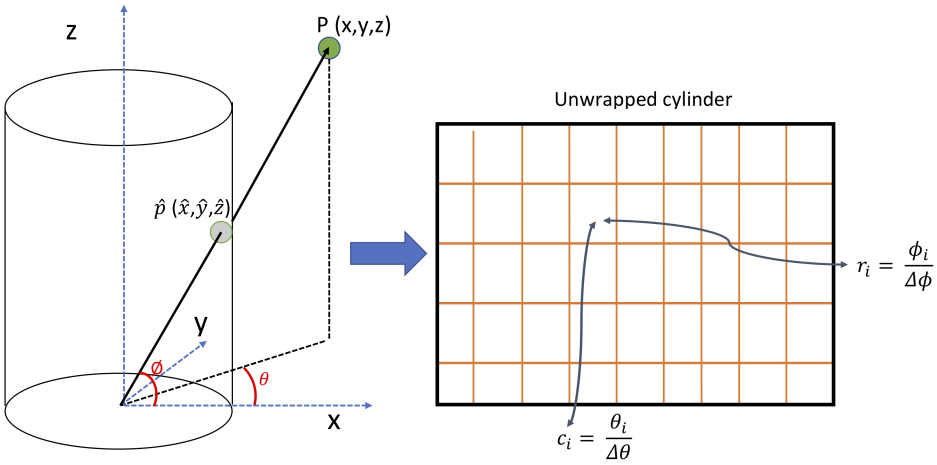


Figure 3 Transformation 3D point cloud to a 2D map

Here, the matrix $\tilde{A} \in \mathbb{R}^{r_X \times r_X}$ is a low-rank operator, with $r_X \ll n$. The incoming snapshot pairs $\mathbf{x}_i$ and $\mathbf{y}_i$ can be large. Thus, we project these snapshots to obtain their low-dimensional representations $\tilde{\mathbf{x}}_i = Q_X^T \mathbf{x}_i$ and $\tilde{\mathbf{y}}_i = Q_Y^T \mathbf{y}_i$, where $Q_Y \in \mathbb{R}^{n \times r_y}$ is an orthonormal basis for the image of Y. The low-dimensional data vectors obtained with multiple such snapshots is given as $\tilde{X} := Q_X^T X$ and $\tilde{Y} = Q_Y^T Y$, respectively. From Eq. (4), we have

$$\tilde{A} := Q_X^T Y X^\dagger Q_X, \tag{5}$$

where $X^\dagger$ denotes the Moore-Penrose pseudoinverse of X. To update the $\tilde{A}$ operator with incoming data, we need to modify the operator in Eq. (5), such that it can be incrementally updated with available data. In the streaming version of DMD we rewrite the DMD operator by substituting $X^\dagger = X^T (X X^T)^\dagger$ in Eq. (5)

$$\tilde{A} = Q_X^T Q_Y K G_X^\dagger, \tag{6}$$

where we define two matrices $K := \tilde{Y}\tilde{X}^T$ and $G_X := \tilde{X}\tilde{X}^T \in \mathbb{R}^{r_X \times r_x}$ for simplicity. In Eq. 6, we will update the operators Q_X, Q_Y, K and G_x when a new snapshot pair becomes available. This enables the linear operator $\tilde{A}$, and thus the DMD modes and eigenvalues to be computed on-the-fly. If the snapshot data contains noise (a high-rank component), we use proper orthogonal decomposition (POD) based compression. The POD compression helps compute Q_X and Q_Y using leading eigenvectors of matrices G_X and G_Y. As seen in G_X, the matrix $G_Y = \tilde{Y}\tilde{Y}^T$. The algorithm for updating the DMD operator with a single pair of snapshot data is as follows:

Algorithm 1 Streaming DMD

procedure STREAMINGDMD
  $e_X \leftarrow x_i - Q_X(Q_X^T xi)$
  $e_Y \leftarrow y_i - Q_Y(Q_Y^T yi)$
  if $||e_X|| > \epsilon$ **then**
    $Q_X(:, end+1) = e_X/||e_X||$
  if $||e_Y|| > \epsilon$ **then**
    $Q_Y(:, end+1) = e_Y/||e_Y||$
  if $rank(Q_X) > r_o$ OR $rank(Q_Y) > r_o$ **then**
    $G_X \leftarrow V_X^T G_X V_X$ ▷ V_X: Leading Eigenvectors of Q_X
    $G_Y \leftarrow V_Y^T G_Y V_Y$ ▷ V_Y: Leading Eigenvectors of Q_Y
    $K \leftarrow V_Y^T K V_X$
    $Q_X \leftarrow Q_X V_X$
    $Q_Y \leftarrow Q_Y V_Y$
  $\tilde{x}_i \leftarrow Q_X^T x_i$
  $\tilde{y}_i \leftarrow Q_Y^T y_i$
  $G_X \leftarrow G_X + \tilde{x}_i \tilde{x}_i^T$
  $G_Y \leftarrow G_Y + \tilde{y}_i \tilde{y}_i^T$
  $K \leftarrow K + \tilde{y}_i \tilde{x}_i^T$

Further details can be found in [7]. We also note that when measurement noise is present, alternative noise-robust formulations of DMD can be used to improve performance [17, 18]. Such considerations are part of ongoing investigations, but are not investigated in the current paper.

We also consider the case where the dynamics embedded in $\tilde{A}$ may be time-varying and hence we use a "forgetting factor" to obtain an adaptive streaming DMD. The forgetting factor enables us to appropriately weight the new incoming snapshot pairs relative to the older snapshot pairs. This weighting factor adds an element of memory to the streaming process that introduces additional versatility to the original streaming DMD formulation. The forgetting factor is denoted by α and satisfies $0 \leq \alpha \leq 1$. In the adaptive streaming DMD algorithm, only steps (16), (17), and (18) differ from the original streaming algorithm presented in *Algorithm 1*. The adaptive streaming DMD algorithm follows:

Algorithm 2 Adaptive Streaming DMD

procedure ADAPTIVESTREAMINGDMD
  $e_X \leftarrow x_i - Q_X(Q_X^T xi)$
  $e_Y \leftarrow y_i - Q_Y(Q_Y^T yi)$
  if $||e_X|| > \epsilon$ **then**
    $Q_X(:, end+1) = e_X/||e_X||$
  if $||e_Y|| > \epsilon$ **then**
    $Q_Y(:, end+1) = e_Y/||e_Y||$
  if $rank(Q_X) > r_o$ OR $rank(Q_Y) > r_o$ **then**
    $G_X \leftarrow V_X^T G_X V_X$ ▷ V_X: Leading Eigenvectors of Q_X
    $G_Y \leftarrow V_Y^T G_Y V_Y$ ▷ V_Y: Leading Eigenvectors of Q_Y
    $K \leftarrow V_Y^T K V_X$
    $Q_X \leftarrow Q_X V_X$
    $Q_Y \leftarrow Q_Y V_Y$
  $\tilde{x}_i \leftarrow Q_X^T x_i$
  $\tilde{y}_i \leftarrow Q_Y^T y_i$
  $G_X \leftarrow G_X(1-\alpha) + \alpha \tilde{x}_i \tilde{x}_i^T$
  $G_Y \leftarrow G_Y(1-\alpha) + \alpha \tilde{y}_i \tilde{y}_i^T$
  $K \leftarrow K(1-\alpha) + \alpha \tilde{y}_i \tilde{x}_i^T$

Each of these two streaming DMD algorithms is effective when the number of snapshots (m) and size of each

snapshot (n) are large. The computational cost of computing DMD modes and eigenvalues using each streaming DMD algorithm is $O(nr^2)$, where $r = \max(r_x, r_y)$. Note that the number of snapshots m does not appear in this scaling, and so the approach is indeed a single-pass low-storage method that is suitable for on-line on-the-fly computations.

V. Results

In this section, we investigate the performance of the streaming DMD algorithm for background-foreground separation. In Section V.A, we discuss in detail the performance of the streaming DMD algorithm and assess the effect of the forgetting factor on video data. The video data is obtained from the "city" category and drive number "2011_09_26_drive_0060" in the KITTI Dataset. In the following section, we test the streaming DMD algorithm and its performance for background and foreground segmentation on LiDAR data streams. The LiDAR data used in section. V.B is from the same corresponding drive in the"city" category, and another scene from the "campus" category with drive number "2011_09_28_drive_0016", both of which are a part of KITTI Dataset. It should be noted that all the LiDAR depth maps have been enhanced for clarity and moving objects have been tagged manually.

A. Background-foreground separation in streaming video

We use the streaming DMD algorithm for background-foreground segmentation using video information from a camera. Once the linear operator $\tilde{A}$ is obtained using either Algorithm 1 or Algorithm 2, eigenvalues provide information on the temporal evolution of the dynamical scene, and hence stationary component and fast moving components are distinguished via eigenvalues. As seen earlier, an image $\mathcal{I}$ can be decomposed into a low-rank component (L) and a sparse component (S). To isolate the low-rank (background) components, we select the DMD modes near the origin ($\omega \approx 0$). Once the DMD modes associated with the background is identified, the sparse component can be directly obtained as $S = \mathcal{I} - L$.

We first study the scene segmentation using standard streaming DMD (Algorithm 1). The video scene used here is from the "city" category in the KITTI Dataset [8]. This particular video is comprised of 75 frames, of which we have shown frame number 6 in Figure 4a. We perform background-foreground segmentation on this video data set using streaming DMD. The background and foreground components have been shown in Figure 4b, 4c. It should be noted that we use all the eigenvalues to perform the scene segmentation; hence, as a by-product we can observe traces of high-frequency components in both the background (Figure 4b) and foreground (Figure 4c). In Figure 4c, the high frequency component leads multiple visible bicycles and the car being blurred out.

Another advantage of using modal decomposition-based methods like DMD is that we can reconstruct scenes using a reduced-order representation. The reduced-order representation is obtained from truncation; where only the dominant DMD modes are used to recreate the scene. This truncation helps in getting rid of eigenvalues related to the high-frequency content. An example of this truncated reconstruction using streaming DMD is shown in Figure 5, here we use only 5 dominant modes to segment background-foreground information from the scene. By comparing the foreground from 75 modes reconstruction (Figure 4c) with 5 mode reconstruction (Figure 5c), we can see that the high-frequency content can be eliminated and the clarity can be improved.

In this work, we use data only from stationary sensors and in cases like this, the DMD based methods can be very powerful—as background foreground information can be obtained using only one mode. Since the one mode selected here will be associated with the background, we can then subtract the obtained background from the original image to obtain foreground content. This can also be understood by observing the eigenvalue plot for the video. In Figure 6, we plot the DMD eigenvalues with $r_x = 50$. In Figure 6, we can see that only the 'red' eigenvalue corresponds to the background mode, while all 'black' eigenvalues combine to give the foreground information. Therefore, a single-mode corresponding to the zero eigenvalue can be used to segment the scene into background and foreground.

We also investigate the effect of the forgetting factor on scene segmentation in the streaming DMD method. To study this, we compare the effect of the forgetting factor on the full-rank decomposition i.e., we do not perform any truncation. In Figure 7, we observe that when the $\alpha = 0.99$ (incoming snapshot pairs are most important) the foreground reconstruction has more high-frequency content compared to the case with $\alpha = 0.5$ (incoming snapshots are weighted equally to past information). In the case with $\alpha = 0$ (incoming snapshots are not considered for updating A matrix), we are not updating the the $\tilde{A}$ matrix with new incoming snapshots, thereby using only initial snapshots to distill background and foreground information. Therefore, a higher forgetting factor corresponds to using more recent snapshot data to update the linear operator $\tilde{A}$. From Figure 7 we see that a decreasing forgetting factor, mostly eliminates the high-frequency content. Even though the forgetting factor has a similar performance as rank truncation, a major benefit of the forgetting factor will be noticed in cases where there is ego-motion with a changing scene. We hypothesize that

(a) Frame no. 6 from vision sensor

(b) Background component of frame no. 6

(c) Foreground component of frame no. 6

Figure 4 Background-foreground segmentation on frame number 6 of city category video data stream; segmentation performed using streaming DMD without truncation. Here, all 75 modes have been used to reconstruct the background scene. The Figure 4b and Figure 4c shows the stationary background and foreground objects respectively, in both we observe traces of high frequency component.

(a) Frame no. 6 from vision sensor

(b) Background component of frame no. 6

(c) Foreground component of frame no. 6

Figure 5 Background-foreground segmentation on frame number 6 of city category video data stream; segmentation performed using streaming DMD with truncation. Here, only 5 modes have been used to reconstruct the background scene. The Figure 5b and Figure 5c shows the background and foreground component with no high frequency component; in comparison to Figure 5b and Figure 5c.

the forgetting factor would be beneficial in such cases, however, ego-motion with changing scenes is a topic we reserve for future investigations.

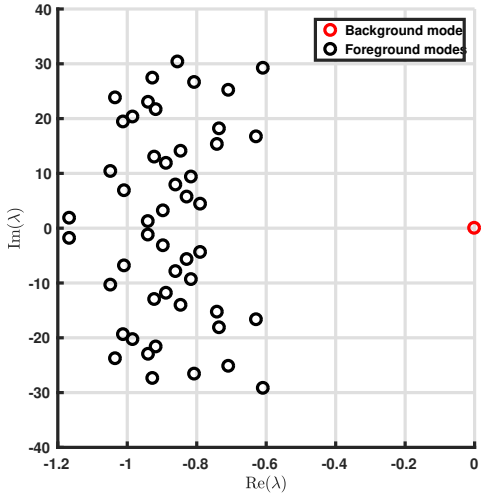


Figure 6 The figure shows the first 50 eigenvalues of the operator $e^{\tilde{A}\delta t}$ obtained by processing the complete video data stream. The eigenvalue corresponding to the stationary mode is shown in red, all other modes correspond to the foreground component in the scene. The existence of a single background mode shows that the background can be reconstructed with a rank 1 matrix.

B. Background-foreground separation from streaming LiDAR data

To enable processing LiDAR point cloud data with the streaming DMD or DMD framework, we convert the point cloud into cylindrical depth map that contain full 360° FOV information using the procedure detailed in Section III. This depth-map is then used for the background-foreground separation with the streaming DMD (both Algorithms 1 & 2).

We show that the LiDAR data streams can also be used to segment background-foreground objects. In Figure 8, we show a depth image of frame number 6 obtained from the LiDAR point cloud; the data is taken from the city category in the KITTI Dataset. The scene is composed of both dynamic and stationary objects. The objects in the scene — a cyclist and a car both of which are moving and have been tagged with a circle and square respectively. All other aspects of the images are stationary and form the background component. Using Algorithm 1, we obtain scene segmentation using 5 dominant modes; both the static and moving objects have been segmented in the scene. The background (stationary) component is shown in Figure 8b and in Figure 8c we show the successful isolation of foreground objects.

In Figure 9a and Figure 10a, we show depth map obtained from a LiDAR point cloud taken from the "campus" category in KITTI Dataset. The power of dynamic mode decomposition-based methods lies in the fact that only the first pair of snapshots are needed to reconstruct the stationary background; and foreground information in the incoming snapshot scenes can be extracted by subtraction. However, this is applicable only in the absence of ego-motion and continuous scene change. This can be observed from Figure 9, where we use only 1 dominant mode to identify the dominant eigenvalues associated with the static background as shown in Figure 9c. In this case, we also set $\alpha = 0$ using Algorithm 2. As mentioned previously, setting $\alpha = 0$ corresponds to the new snapshot pair is not being used in updating the linear operator $\tilde{A}$. Therefore, we can segment background and foreground information only using a single snapshot pair for settings where the scene is not continuously changing.

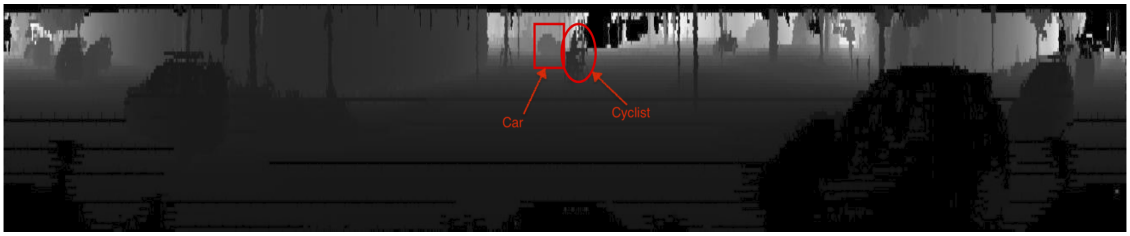
We also study the combined effect of truncation with forgetting factor $\alpha = 0.5$. This setting is very useful for real-world applications, where we want to compute background-foreground information from reduced-rank operators, while weighting the incoming snapshot. In Figure 11, we illustrate the effect of combining truncation and snapshot weighting on the LiDAR data from the city category. Here, the reduced-rank operator $\tilde{A}$ is of rank 50. It can be seen that with both the truncation and forgetting factor, the foreground objects (see Figure 11d) have been distilled from the background (see Figure 11c). Even though this has just been demonstrated for one scene only, we observe similar results when this method is applied to other scenes as well.

(a) $\alpha = 0.99$

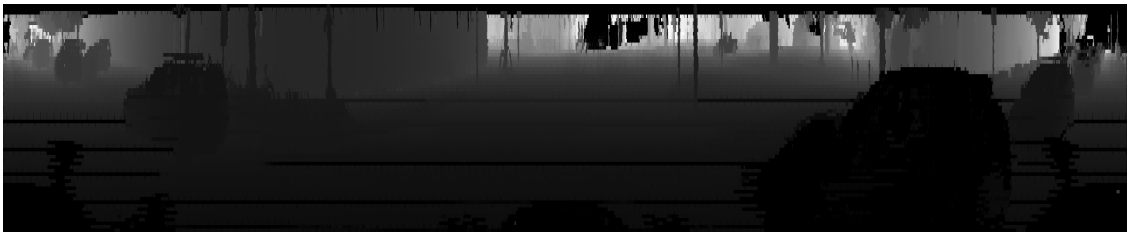
(b) $\alpha = 0.5$

(c) $\alpha = 0$

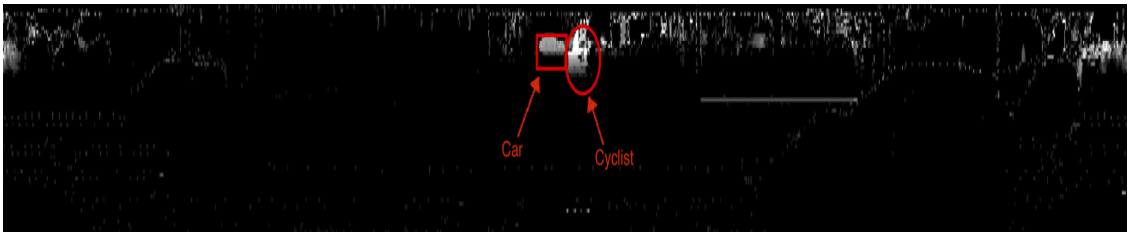
Figure 7 The figure shows the effect of forgetting factor in segmenting foreground objects on video data streams. A high forgetting factor retains high frequency content. Similarly, $\alpha = 0$ implies that the foreground has been obtained processing only initial snapshot pair and without processing additional incoming snapshots.



(a) Actual scene with tags



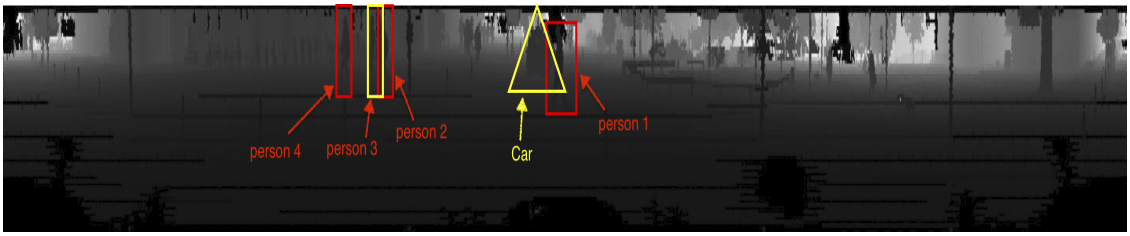
(b) Background component



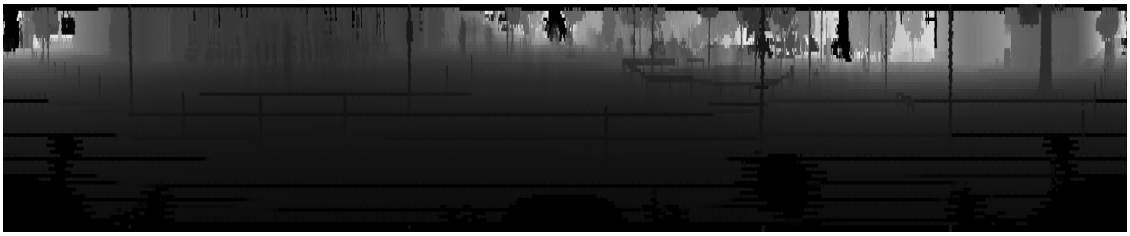
(c) Foreground components

Figure 8 Depth map obtained from scan 3 of city category LiDAR point cloud. Background-foreground segmentation on LiDAR depth map using streaming DMD. Only 5 mode have been used to perform background-foreground segmentation. The full scene is shown in Figure 8a; here the cyclist (tagged with a circle) and car (tagged with a square) are the only moving objects. The static background information is shown in (Figure 8b). The isolated dynamic foreground objects the cyclist and car are shown in Figure 8c.

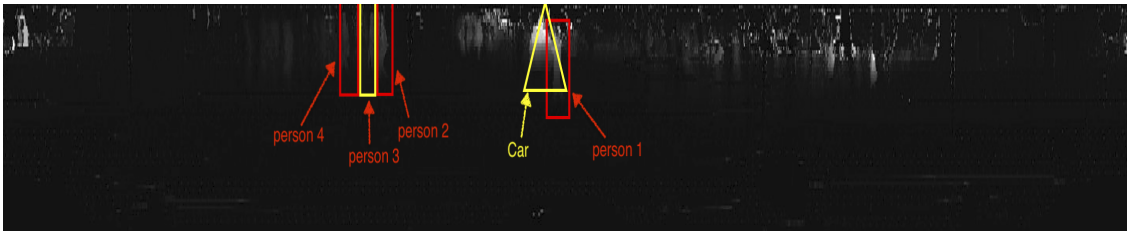
(a) Depth map obtained from scan 1 of LiDAR point cloud



(b) Depth map obtained from scan 1 of LiDAR point cloud, shown here with tags on moving objects



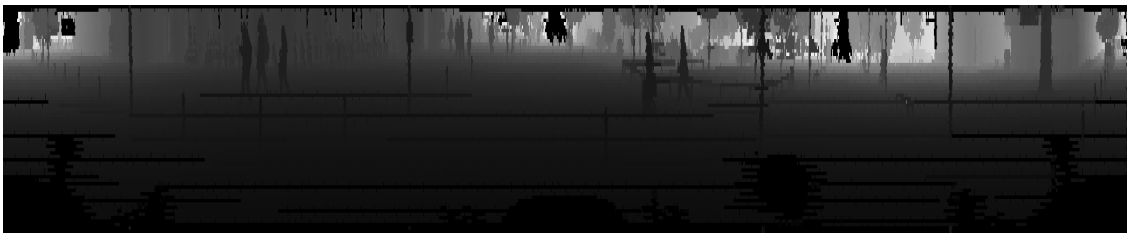
(c) Background component of scan 1



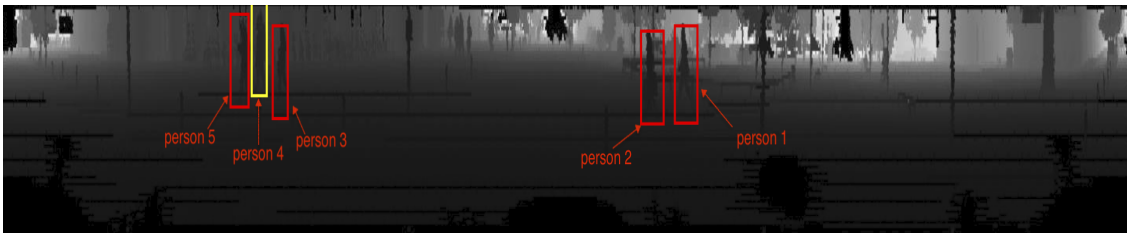
(d) Foreground component of scan 1

Figure 9 Depth map obtained from scan 1 of campus category LiDAR point cloud. Background-foreground segmentation on LiDAR depth map using streaming DMD. The figures in Figure 9a and Figure 9b are from the same scene, the tags have been manually added in the latter to show moving objects of interest. We reconstruct the background using only first 2 snapshot pairs as shown in 9c. In addition to the background, the foreground component is also shown in Figure 9d

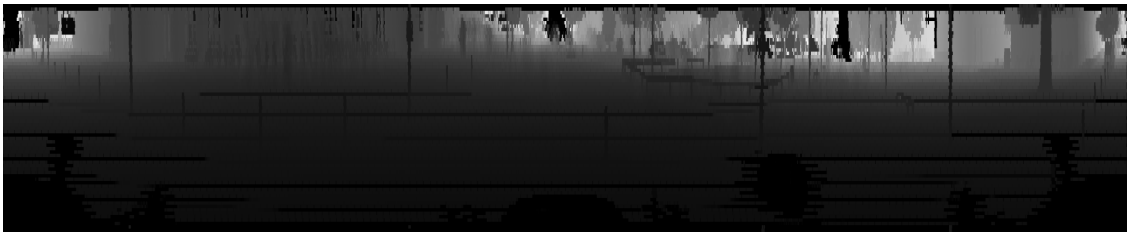
.



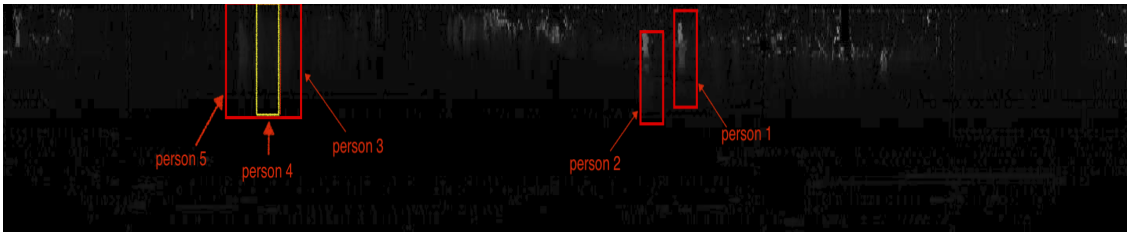
(a) Depth map obtained from scan 96 of LiDAR point cloud



(b) Depth map obtained from scan 96 of LiDAR point cloud, shown here with tags on moving objects

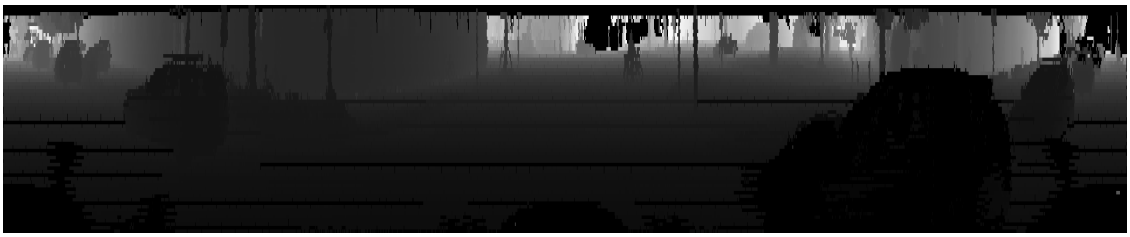


(c) Background component of scan 96



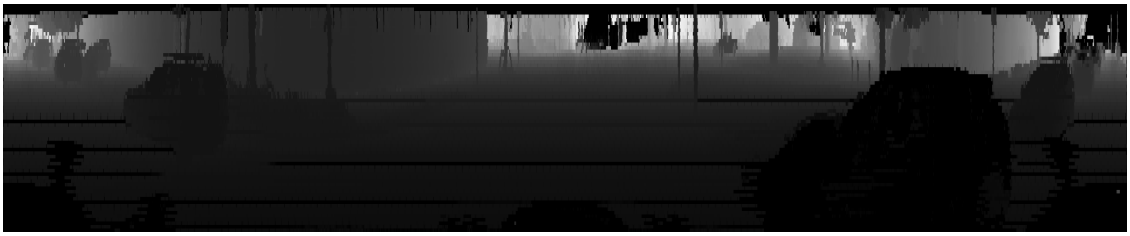
(d) Foreground component of scan 1

Figure 10 Depth map obtained from scan 96 **of campus category LiDAR point cloud. Background-foreground segmentation on LiDAR depth map using streaming DMD. The figures in Figure 10a and Figure 10b are from the same scene, the tags have been manually added in the latter to show moving objects of interest. We reconstruct the background using only first** 2 **snapshot pairs as shown in 10c. In addition to the background, the foreground component is also shown in Figure 10d**

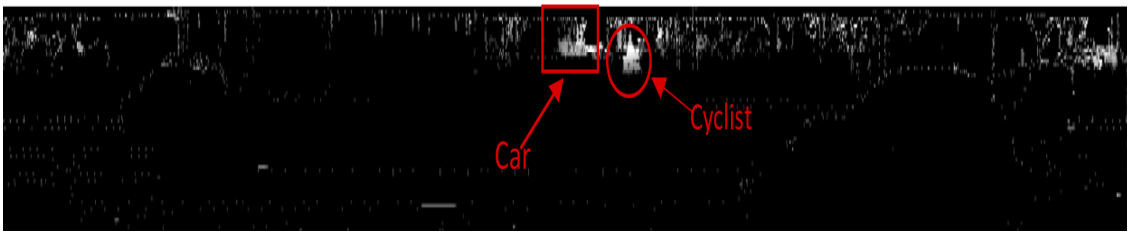


(a) Depth map obtained from scan 17 of LiDAR point cloud

(b) Depth map obtained from scan 17 of LiDAR point cloud, shown here with tags on moving objects



(c) Background component of scan 17



(d) Foreground component of scan 17

Figure 11 Depth map obtained from scan 17 of city category LiDAR point cloud. Background-foreground segmentation on LiDAR depth map using adaptive streaming DMD. The figures in Figure 11a and Figure 11b are from the same scene, the tags have been manually added in the latter to show moving objects of interest. We reconstruct the background using reduced-rank operator of rank 50 snapshot with $\alpha = 0.5$ as shown in 11c. In addition to the background, the foreground component is also shown in Figure 11d

VI. Conclusions

We demonstrate the ability of the streaming DMD approach in foreground-background scene segmentation from both video and LiDAR sensors. Here, moving foreground objects can be isolated from the stationary backgrounds in an "on-line" setting, when a new frame becomes available. We show that a lower rank approximation of the scene can be obtained by truncation; the truncation process can also be used to eliminate spurious high-frequency components in the scene.

The forgetting factor has also been shown to be effective in weighting the new incoming snapshot data and behaves similar to the reduced-rank truncation. Even though we study only the case with static sensors, a natural extension of this work is to apply the streaming DMD algorithms to moving platforms and changing scenes. The forgetting factor used in SDMD will be especially helpful in cases with ego-motion and changing scenes. This work can be further extended by developing tracking and filtering algorithms to track dynamical objects in the scene.

Under certain conditions, the streaming approach can reconstruct the complete background component of the scene using only a few snapshot pairs. This is especially useful when data is only available intermittently or after long intervals. Therefore, streaming DMD is an effective and apt method for background-foreground separation applications on registered video and LiDAR data feeds. This is specifically effective in autonomous systems that receive streaming information from on-board sensors and need to perform tasks like object detection, tracking and classification.

VII. Acknowledgements

This material is based upon work supported by Honeywell Aerospace and MnDRIVE. MSH Thanks Dr. Matthew O. Williams for initial discussions related to streaming DMD for video processing applications.

References

[1] Li, B., Zhang, T., and Xia, T., "Vehicle Detection from 3D Lidar Using Fully Convolutional Network," *Robotics: Science and systems conference*, 2016. doi:10.15607/RSS.2016.XII.042.

[2] Yang, B., Luo, W., and Urtasun, R., "PIXOR: Real-time 3D Object Detection from Point Clouds," *Proccedings of IEEE conference on Computer Vision*, 2018.

[3] Candès, E. J., Li, X., Ma, Y., and Wright, J., "Robust principal component analysis?" *Journal of the ACM*, 2011. doi: 10.1145/1970392.1970395.

[4] Grosek, J., and Kutz, J. N., "Dynamic mode decomposition for real-time background/foreground separation in video," *arXiv*, 2014, p. 14.

[5] Schmid, P., "Dynamic mode decomposition of numerical and experimental data," *J. Fluid Mech*, Vol. 656, 2010, pp. 5–28. doi:10.1017/s0022112010001217.

[6] Tu, J. H., Rowley, C. W., Luchtenburg, D. M., Brunton, S. L., and Kutz, J. N., "Dynamic Mode Decomposition : Theory and Applications," *Journal of Computational Dynamics*, , No. September, 2013, pp. 1–141. doi:10.3934/jcd.2014.1.391.

[7] Hemati, M. S., Williams, M. O., and Rowley, C. W., "Dynamic mode decomposition for large and streaming datasets," *Physics of Fluids*, Vol. 26, No. 11, 2014. doi:10.1063/1.4901016.

[8] Geiger, A., Lenz, P., Stiller, C., and Urtasun, R., "Vision meets Robotics: The KITTI Dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[9] Luber, M., Arras, K., Plagemann, C., and Burgard, W., "Classifying Dynamic Objects: An Unsupervised Learning Approach," *Robotics: Science and Systems IV, Zurich*, 2008. doi:10.15607/RSS.2008.IV.035.

[10] Fergus, R., Perona, P., and Zisserman, A., "Object class recognition by unsupervised scale-invariant learning," *Computer Vision and Pattern Recognition*, 2003. doi:10.1109/cvpr.2003.1211479.

[11] Kotsiantis, S. B., "Supervised machine learning: A review of classification techniques," *Informatica 31*, 2007.

[12] Himmelsbach, M., Müller, A., Luettel, T., and Wuensche, H.-J., "LIDAR-based 3D object perception," *Proceedings of 1st International Workshop on Cognition for Technical Systems, Munich*, 2008.

[13] Premebida, C., Monteiro, G., Nunes, U., and Peixoto, P., "A Lidar and Vision-based Approach for Pedestrian and Vehicle Detection and Tracking," *Intelligent Transportation Systems Conference*, 2007.

[14] Rao Gangineni, S., Reddy Nalla, H., Fathollahzadeh, S., and Teymourian, K., "Grand Challenge: Real-Time Object Recognition from Streaming LiDAR Point Cloud Data," *Proceedings of the* 13^{th} *ACM international conference on DEBS, Darmstadt, Germany*, 2019. doi:10.1145/3328905.3330297.

[15] Taira, K., Brunton, S. L., Dawson, S. T. M., Rowley, C. W., Colonius, T., McKeon, B. J., Schmidt, O. T., Gordeyev, S., Theofilis, V., and Ukeiley, L. S., "Modal Analysis of Fluid Flows: An Overview," *AIAA Journal*, 2017.

[16] Taira, K., Hemati, M. S., Brunton, S. L., Sun, Y., Duraisamy, K., Bagheri, S., Dawson, S. T. M., and Yeh, C.-A., "Modal Analysis of Fluid Flows: An Overview," *AIAA Journal*, 2017.

[17] Hemati, M. S., Rowley, C. W., Deem, E. A., and Cattafesta, L. N., "De-biasing the dynamic mode decomposition for applied Koopman spectral analysis of noisy datasets," *Theoretical and Computational Fluid Dynamics*, Vol. 31, No. 4, 2017, pp. 349–368. doi:10.1007/s00162-017-0432-2.

[18] Hemati, M. S., Deem, E. A., Williams, M. O., Rowley, C. W., and Cattafesta, L. N., "Improving Separation Control with Noise-Robust Variants of Dynamic Mode Decomposition," *AIAA Paper 2016-1103*, 2016.