Locating events using independent component analysis and Gaussian mixture models

Heather K. Hardeman-Vooys*, Matt McDonald†, and Michael P. Lamoureux *

ABSTRACT

Inspired by the work of Shamsa and Paydayesh (2019) who used Gaussian mixture models and independent component analysis to analyze microseismic data, we apply the methodology to data collected using a distributed acoustic sensor in order to detect a vehicle driving along a DAS system. We introduce Gaussian mixture models and independent component analysis. Then, we provide two examples where we calculate two independent components, the vehicle's signal and the noise, before training a Gaussian mixture model to detect the signal. We consider two methods of training the Gaussian mixture model and compare the results. Finally, we conclude.

INTRODUCTION

Distributed acoustic sensors have practical use in security and monitoring (Hartog, 2017). Detecting events in the DAS-acquired data becomes a major issue, especially since such data can cover time-intervals spanning hours. It thus becomes essential to be able to detect events within terabytes of data. In this paper, we offer an analysis of one technique for locating these events inspired by the authors of Shamsa and Paydayesh (2019): Gaussian mixture models.

In Shamsa and Paydayesh (2019), the authors employ independent component analysis (ICA) with a Gaussian mixture model (GMM) to detect events in microseismic data. They utilize ICA to enhance the feature they wish to detect in the microseismic signal with the GMM. We apply a similar idea to a data set acquired using a DAS system installed next to a road, and used to detect a vehicle driving along a road. The GMM extracts the foreground of the data which contains the vehicle's signal. We will refer to the detector as the vehicle detector. We compare the use of ICA beforehand in a variety of different methods.

Independent component analysis was created as an answer to the *cocktail party problem*, where the objective is to differentiate between two speakers based on several microphones collecting acoustic information. Given that a DAS system only possesses one receiver, we use several data sets of the same region of the fibre to act as the additional "microphones" in this case. After a brief introduction to the methodology of the chapter, Gaussian mixture models, and independent component analysis, we begin by applying an ICA to segments of multiple data sets to enhance the vehicle signal over the noise. In the first half of the chapter, we chose the multiple data sets to contain a majority of data with the vehicle's signal and a minority which contained only noise. Independent component analysis gives two independent components: one for the vehicle signal and one for the noise. Afterwards, we then train a GMM on each set of independent components separately

*University of Calgary, Department of Mathematics
†Fotech Solutions

before applying the two GMMs to a test data set. We also provide the results of training the GMM on a training data set. We compare the effectiveness of each method by calculating the percentage of overlap the detector produces for locating the vehicle's signal in the data. For the first case, the GMM without an ICA applied beforehand performed better, achieving between approximately 91% and 93% overlap of the signal as opposed to the GMM with an ICA, which achieved between approximately 20% and 43% overlap of the vehicle signal in the test data. This is likely due the fact that ICA cannot order the independent components, i.e. the vehicle signal may not be assigned to the same component for each segment (Hyvärinen and Oja, 2000).

In the second example, we apply an ICA to a larger segment of the multiple data sets and then train the GMM on a sliding window (the same size as the segments from the first example) across the larger independent component results. In this case, the vehicle signal and noise were delegated to two separate independent components unlike the overlap we saw in the previous case. We window a GMM across both independent components, and compare the results of the GMM without an ICA applied. The GMM taught using the noisy independent component achieved a overlap between approximately 59% and 80% whereas the GMM trained on the vehicle independent component earned a overlap between approximately 90% and 93%. The GMM taught on the training set acquired a overlap between approximately 97% and 98%. The increase in overlap between the two cases likely stems from the fact that more frames were employed to train the GMMs. This improved the results of the GMM taught on the training data. The consistent signal for the independent components also factor in the significant increase in overlap.

The GMM trained on a training set performed better than the GMMs trained on the independent components in every case; however, the GMMs trained on the independent components often provided a more precise detection of the event unlike the GMM taught using the training data. The success of the Gaussian mixture model in detecting the vehicle's signal across both methods still shows that this path of event detection holds promise. In the future, using more specific data sets to enhance the vehicle signal with the independent component analysis is worth investigation. Employing fewer data sets by choosing ones which contain no vehicle signal and a single data set which contains the vehicle signal may improve the results of the ICA to extract the vehicle signal from the noise. This works on the idea that the noise should be similar across the DAS system's monitoring of the road.

METHODOLOGY

In Shamsa and Paydayesh (2019), the authors outline a method which uses a Gaussian mixture model (GMM) and independent component analysis (ICA) to separate the signal from the noise in microseismic data. They found that applying an independent component analysis from Guoshen (2012) to the data prior helped with detection and used this method on data being acquired in real time. The authors saw improvement when more data was fed through the ICA and GMM as it allowed the ICA to better distinguish between noise and the signal; however, it was possible to provide too much data and cause the error from the ICA to grow too large.

Following a similar vein, we utilize the `fastICA` function described in Gävert et al.

(2005) and employ it with regards to the foreground detector from MATLAB following the example from MATLAB (2019). In the MATLAB example, the Gaussian mixture model differentiates the foreground from the background to identify cars in video data. The first few frames of the video contain no images of cars in order to define the background for the Gaussian mixture model. As the cars arrive in the frames, the GMM distinguishes them from the background. We apply this methodology to data of a vehicle driving along a road, acquired using a distributed acoustic sensor. While the example's application is to images which contain cars, we use the foreground detector on DAS-acquired data to indicate the signal of a vehicle. We refer to it as the vehicle detector in the applications later in this chapter. We also administer additional morphological cleaning to the data than what is found in the MATLAB example.

In the following sections, we provide a brief study of Gaussian Mixture Models and their use in foreground detection. We then examine the algorithm behind independent component analysis and its ability to distinguish between a signal and noise in data. Afterwards, we apply the methodology described in Shamsa and Paydayesh (2019) to DAS-acquired data of a vehicle driving along a road.

GAUSSIAN MIXTURE MODELS

Traffic monitoring often employs Gaussian mixture models for foreground detection where the cars lie in the foreground. We follow an example from MATLAB that applies GMMs to data from a video data of cars driving down a road (MATLAB, 2019). It bases the foreground detector functions on the Gaussian mixture models described in Stauffer and Grimson (1999) and KaewTraKulPong and Bowden (2001). Stauffer and Grimson sought to create a foreground detector which could deal with arbitrary variations in the background; in particular, those found in video data such as the shadows from the sun's movement, moving tree branches, etc. Their answer to address these changes in the background of video data involves describing each pixel of an image via some number of Gaussians distributions, often ranging between three and five.

Distributed acoustic sensors do not produce video data; however, they are used in monitoring projects over large periods of time. Segmenting the data in the form of a video makes the data more manageable. It also lowers the computational costs of applying processing techniques. Employing a Gaussian mixture model on DAS-acquired data enables the distinction between the background noise and the foreground where events such as walking or digging may occur. Despite the lack of visual concerns such as shadows due to lighting or moving tree branches, data often contains noise or other information which can overpower events in the data that we wish to detect. The GMM treats the noise as the background and produces a foreground where the events reside. In the Examples section, a DAS system is employed to detect a vehicle driving by the road. The goal of the GMM is to distinguish between the vehicle's signal and the background noise.

Stauffer and Grimson (1999) follow two basic steps in their proposed GMM:

1. Model the values of a specific pixel as a mixture of Gaussian distributions.
2. Assign the pixel to the foreground or background based on persistence and variance

of each Gaussian distribution in the mixture.

Each pixel in the image undergoes a "pixel process." At any point in time, the pixel history is known, i.e.

$$\{X_1, ..., X_t\} = \{\mathbf{I}(x_0, y_0, i) : 1 \leq i \leq t\} \tag{1}$$

where X_i is the pixel value and $\mathbf{I}$ is the image sequence. The recent history of each pixel is modeled by a mixture of K Gaussian distributions. Let $w_{i,t}$ be the weight estimate of the ith Gaussian in the mixture at time t, $\mu_{i,t}$ be the mean value of the ith Gaussian in the mixture at time t, and $\Sigma_{i,t} = \sigma_i^2 I$ is the covariance matrix of the ith Gaussian in the mixture at time t. We calculate the probability of the current pixel value occurring via the equation:

$$P(X_t) = \sum_{i=1}^{K} w_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \tag{2}$$

where the Gaussian probability density function is given as

$$\eta(X_t, \mu_{i,t}, \Sigma_{i,t}) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma_{i,t}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(X_t - \mu_{i,t})^T \Sigma_{i,t}^{-1}(X_t - \mu_{i,t})\right). \tag{3}$$

To maximize the likelihood of the observed data, the authors execute an online K-means approximation since an exact expectation-maximization (EM) algorithm would be computationally expensive. Each new pixel is compared to the existing K Gaussian distributions until we find one within 2.5 standard deviations of a distribution. When this occurs, we have a match. If the pixel is not within 2.5 standard deviations of any of the K Gaussian distributions, then we replace the least probable distribution with the current pixel value as well as its mean value, an initially high variance, and the low prior weight. We update the weight using the following update step:

$$w_{i,t} = (1 - \alpha) w_{i,t-1} + \alpha M_{i,t}, \tag{4}$$

where α is the learning parameter and

$$M_{i,t} = \begin{cases} 1 & \text{for a model which matched the current pixel value;} \\ 0 & \text{for remaining models.} \end{cases} \tag{5}$$

The parameters μ and σ are updated as follows:

i. For all unmatched Gaussian distributions, μ and σ remain the same;

ii. For the matched distribution,

$$\mu_{i,t} = (1 - \rho)\mu_{i,t-1} + \rho X_t; \tag{6}$$

$$\sigma_{i,t}^2 = (1 - \rho)\sigma_{i,t-1}^2 + \rho(X_t - \mu_{i,t})^T(X_t - \mu_{i,t}), \tag{7}$$

where

$$\rho = \alpha \eta(X_t | \mu_{i,t}, \sigma_{i,t}). \tag{8}$$

Once each pixel is assigned to a Gaussian distribution in the mixture, we order the Gaussian distributions using the parameter w/σ which puts the most probable background distributions at the top of the list and the least probable background distributions at the bottom to eventually be replaced by new distributions. Let

$$B = \underset{b}{\operatorname{argmin}} \Big(\sum_{k=1}^{b} w_k > T \Big), \tag{9}$$

where T is the measure of the minimum portion of the data for which the background should be accounting. We choose the first B Gaussian distributions in our list to model the background of the data.

We use a two-pass, connected component algorithm to segment labeled foreground pixels into regions. In Stauffer and Grimson (1999), the authors employ a linearly predictive multiple hypothesis tracking algorithm to correlate connected component between frames of data. The algorithm utilizes the position and size from the previous frame to make its judgment about the next. Matching the models to connected components typically requires comparing each model against the current pool of connected components.

In KaewTraKulPong and Bowden (2001), the authors improved the work in Stauffer and Grimson (1999) by adjusting the update step for the weights w, and the parameters μ and σ, as well as including a threshold for shadow detection. Instead of using a K-means approximation, KaewTraKulPong and Bowden implemented an EM-algorithm for the first L windows using the following update equations:

$$w_{i,t+1} = w_{i,t} + \frac{1}{t+1}\Big(M_{i,t+1} - w_{i,t}\Big); \tag{10}$$

$$\tag{11}$$

$$\mu_{i,t+1} = \mu_{i,t} + \frac{M_{i,t+1}}{\sum_{k=1}^{t+1} M_{i,k}}\Big(X_{t+1} - \mu_{i,t}\Big); \tag{12}$$

$$\tag{13}$$

$$\Sigma_{i,t+1} = \Sigma_{i,t} + \frac{M_{i,t+1}}{\sum_{k=1}^{t+1} M_{i,k}}\Big((X_{t+1} - \mu_{i,t})(X_{t+1} - \mu_{i,t})^T - \Sigma_{i,t}\Big), \tag{14}$$

where w_i is the k-th Gaussian component and the parameter $1/\alpha$ defines the time constant. Then, they apply a L-recent windows update using the following equations:

$$w_{i,t+1} = w_{i,t} + \frac{1}{L}\Big(M_{i,t+1} - \omega_{i,t}\Big); \tag{15}$$

$$\tag{16}$$

$$\mu_{i,t+1} = \mu_{i,t} + \frac{1}{L}\Big(\frac{M_{i,t+1}X_{t+1}}{\omega_{i,t+1}} - \mu_{i,t}\Big); \tag{17}$$

$$\tag{18}$$

$$\Sigma_{i,t+1} = \Sigma_{i,t} + \frac{1}{L}\Big(\frac{M_{i,t+1}}{\omega_{i,t+1}}(X_{t+1} - \mu_{i,t})(X_{t+1} - \mu_{i,t})^T - \Sigma_{i,t}\Big). \tag{19}$$

They achieved the shadow detection in KaewTraKulPong and Bowden (2001) using a position vector at the RGB mean of the pixel background, E, an expected chromaticity distortion, d, and a brightness threshold, τ. We calculate the brightness distortion a and the colour distortion c:

$$a = \underset{z}{\operatorname{argmin}}(X_t - zE)^2; \tag{20}$$

$$c = \|X_t - aE\|_2 , \tag{21}$$

for an observed pixel value, X_t. The authors define a moving shadow if the brightness distortion a is within 2.5 standard deviations and $\tau < c < 1$.

While this Gaussian mixture model is geared toward video data, we adapt it work on DAS-acquired data in the Examples section to detect a vehicle driving along a fibre-optic cable receiver.

INDEPENDENT COMPONENT ANALYSIS

To help the GMM differentiate between the vehicle's signal and the background noise of the data, we utilize independent component analysis for the purposes of feature extraction. Independent component analysis offers a solution to the cocktail problem which asks the solver to distinguish between two individuals speaking simultaneous given two recordings from two microphones at different locations in the room.

Mathematically, we can write this problem as

$$v_1(t) = a_{11}s_1(t) + a_{12}s_2(t); \tag{22}$$

$$v_2(t) = a_{21}s_1(t) + a_{22}s_2(t), \tag{23}$$

where v_1 and v_2 are the recordings from the microphones, a_{ij} for $1 \leq i, j \leq 2$ are the parameters describing the distance between each speaker and the microphone, and s_1 and s_2 are the speech signals from each speaker respectively. This problem can be written as the linear algebra equation

$$\mathbf{v} = A\mathbf{s}. \tag{24}$$

Independent component analysis provides an estimate of the independent components $\mathbf{s}$ based solely on information from $\mathbf{v}$. It does this by solving for an approximation of A which it then computes the inverse of A to solve

$$\mathbf{s} = A^{-1}\mathbf{v}. \tag{25}$$

Before the ICA algorithm is applied, the data $\mathbf{v}$ must be whitened (Hyvärinen and Oja, 2000). This process involves transforming $\mathbf{v}$ into a new vector whose components are uncorrelated and the variance of these components equals 1, i.e., $\mathbf{x} = M\mathbf{v}$ where $\mathbf{x}$ is the correlation matrix satisfying $E\{\mathbf{x}\mathbf{x}^T\} = 1$. The method we employ later for the ICA algorithm uses principal component analysis to whiten $\mathbf{x}$.

We implement a fixed-point ICA algorithm for estimating multiple independent components based on the work in Hyvärinen (1999). Hyvärinen (1999) defines the algorithm as follows: Assume we have a sample of the prewhitened random vector $\mathbf{x}$:

1. Take a random initial vector $\mathbf{w}(0)$ where $\|\mathbf{w}\| = 1$. Let $k = 1$.

2. Let
$$\mathbf{w}(k) = E\{\mathbf{x}(\mathbf{w}(k-1)^T\mathbf{w}(k-1))^3\} - 3\mathbf{w}(k-1). \tag{26}$$
Estimate the expectation E using a large sample of vectors $\mathbf{x}$.

3. (a) Update $\mathbf{w}(k)$ to the quantity $\mathbf{w}(k) - \overline{\mathbf{B}}\overline{\mathbf{B}}^T\mathbf{w}(k)$ and replace $\mathbf{w}(k)$ with the update. Symbolically,
$$[\text{Updated}]\ \mathbf{w}(k) := \mathbf{w}(k) - \overline{\mathbf{B}}\overline{\mathbf{B}}^T\mathbf{w}(k). \tag{27}$$
(b) Compute
$$\frac{\mathbf{w}(k)}{\|\mathbf{w}(k)\|}. \tag{28}$$

4. For $|\mathbf{w}(k)^T\mathbf{w}(k-1)|$ not close to 1, let $k = k + 1$ and return to 2. Otherwise, output the vector $\mathbf{w}(k)$ into a column of the orthogonal mixing matrix B.

The parameter $\overline{\mathbf{B}}$ is the matrix whose columns are taken from the previously found columns of the orthogonal mixing matrix $\mathbf{B}$, where $\mathbf{s} = \mathbf{B}^T\mathbf{x}$. The columns of $\mathbf{B}$ are the independent components computed by the algorithm. Once the solution $\mathbf{w}(k)$ reaches the basins of attractions of one of the fixed points, we drop step 3(a) in order to prevent an estimation error from building in $\overline{\mathbf{B}}$.

We employ MATLAB code written by Hugo Gävert, Jarmo Hurri, Jaakko Särelä, and Aapo Hyvärinen to apply the FastICA algorithm to the examples in the Examples section (Gävert et al., 2005). The authors based their `fastICA` function on the work of Hyvärinen (1999) and Hyvärinen and Oja (2000).

A significant factor to address with respect to independent component analysis is that the independent components must be non-Gaussian (Hyvärinen and Oja, 2000). If both components are Gaussian, then the matrix A cannot be estimated. While the combination of independent component analysis with a mixture model which estimates pixel values based on Gaussians seems problematic, the foreground detector in a GMM employs a mixture of Gaussians to estimate each pixel which does not mean each pixel itself is Gaussian. We later apply the GMM to the independent components generated by the ICA so the foreground detector using Gaussian distributions to estimate the points in the independent components. Because we apply the ICA to the data first and then the GMM to the independent components, this will not affect how the ICA works with the data.

EXAMPLES

For the following sections, we examine different methods for applying the GMM and ICA to a data set of a vehicle driving along a road next to a fibre-optic sensor. Our attention resides on a portion of the road that is approximately 65m in length, and focus on it for about 70 seconds. We start with the application of a Gaussian mixture model to a single data set. We consider the results of administering an ICA to segments of multiple data

sets of the same portion of road. Finally, we apply an ICA to multiple large data sets and then apply the GMM to the resulting independent components. Recall that we refer to the foreground detector from the literature as a vehicle detector in this paper, given that we use the GMM to detect the vehicle's signal.

Figure 1 depicts the data we use to train the GMM in the following examples when we do not apply an ICA first. A Fotech DAS system collected the data of a vehicle driving on a road with a fibre-optic cable laid beside it. The data is approximately 135 seconds in length and considers a 65m stretch of fibre. It shows a vehicle driving along the fibre-optic cable between 90 and 105 seconds.

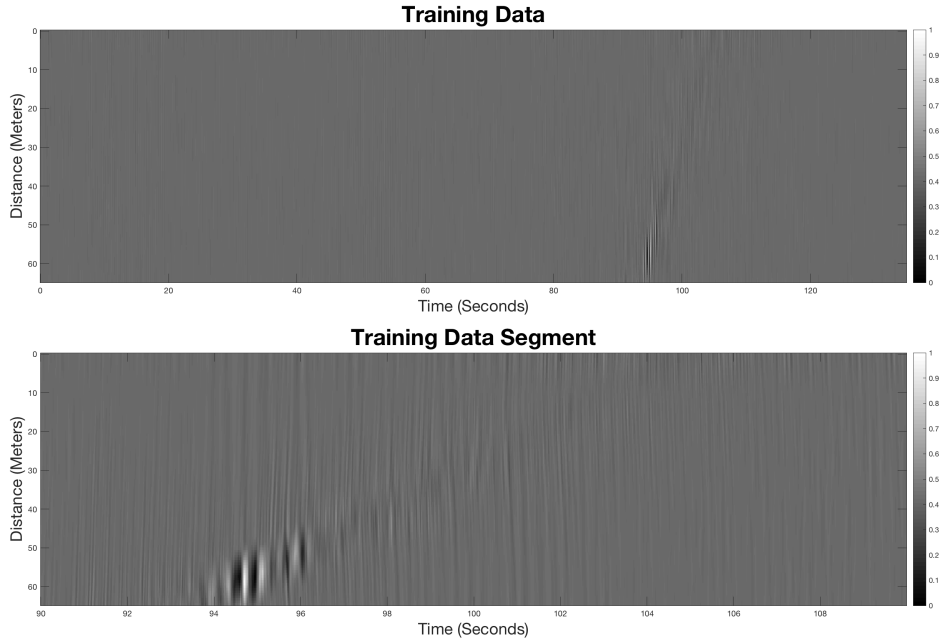

FIG. 1: (Top) The training data set: A 65 meter section of fibre with a vehicle driving along a road next to a DAS system. The vehicle starts at 90 seconds and goes to 105 seconds. (Bottom) We show a 90 second segment of the training set seen above. We focus on the data between 65 seconds to 110 seconds where the vehicle is located in the training set.

In order to offer a comparison of the effectiveness of each method's vehicle detector, we apply the resulting vehicle detectors in each section to a test data set; c.f. Figure 2. The figure shows the data set of a vehicle driving along a road beside a fibre-optic cable. It starts with no vehicle present. After 10 seconds, the vehicle appears and drives along the 65m of fibre for 60 seconds before the data ends. In each of the following sections, we observe how well the vehicle detectors detect the vehicle in the test data set from Figure 2.

There is only one DAS system in this setup, which means there is only one microphone for the independent component analysis in each example. Given that the background noise should be similar over the long period of time, we generate more "microphones" by considering data sets acquired at different times for the same section of fibre. While this may provide us with a good estimation of the noise, it may cause issues with how well the ICA

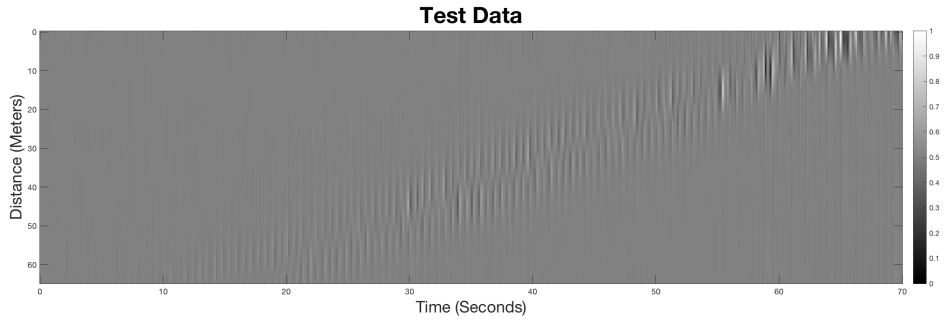

FIG. 2: The test data set: A 65 meter section of fibre with a vehicle driving along a road next to a DAS system. The vehicle starts at 10 seconds and goes to 70 seconds.

depicts the signal. A problem may arise if we use data with a variety of vehicle signals, such as the vehicle driving from 0m to 65m and the vehicle driving from 65m to 0m. We keep this in mind as we consider the following examples.

Case 1: GMM applied to independent components generated from segments of multiple data sets

The authors of Shamsa and Paydayesh (2019) found that segmenting the data sets before performing an independent component analysis to distinguish the microseismic event signal from the noise improved their results as it allowed less room for accumulating error. We begin in a similar fashion by applying an independent component analysis on segments of 20 data sets of the vehicle driving along the same 65m length of road next to a fibre-optic cable. Each segment contains 5 seconds and the entire 65 meters of fibre. We remove every 10th shot from the data to form a new data set. Then, we segment the new data. For the independent component analysis, we only consider two independent components: the signal of the vehicle and noise. In this case, we use segments from 20 data sets which show the data from the same segment of fibre in time to find the two independent components. We then use the GMM on the independent components containing the signal of the vehicle for each segment of the data.

Let us consider the values we chose for three parameters of the vehicle detector: the number of Gaussians modes, the number of training frames, and the learning rate. For this case, we decided to use three Gaussian modes. Given that our data is short in time, we only get 27 segments from the each data set. Each independent component has 27 frames. We set the number of training frames for the vehicle detector to be 10. We use approximately 1/3 of the frames to train the vehicle detector. Finally, we chose the learning rate to be 0.0001. The learning rate affects how fast the model adapts to change in the parameters.

Before we examine the results from the vehicle detectors taught on the independent components, we train on the training set to get some idea of how the GMM should perform visually. Given that the training set is the same size as the data sets we used to determine the independent components, we set the number of training frames to be 10 in this case as well. This is approximately $1/3$ of the frames from the training set, keeping in line with the percentage of frames used for the vehicle detector of the independent components.

After the detector is trained, the MATLAB example applies a morphological cleaning to the proposed masks of the vehicle for each frame. For our experiment, we apply an additional cleaning to the vehicle mask which bridges any 0-valued pixels in the mask which are surrounded by two or more 1-valued pixels as well as determining the value of a pixel by taking into account the majority of values around said pixel and forcing it to match. We wish to detect the vehicle the entire time so bridging and filling holes in the mask is not going to be an issue. Figure 3 presents the results of the vehicle detector trained on the training data set. The top figure shows the results after the initial cleaning and the bottom figure displaces the results after the second cleaning.

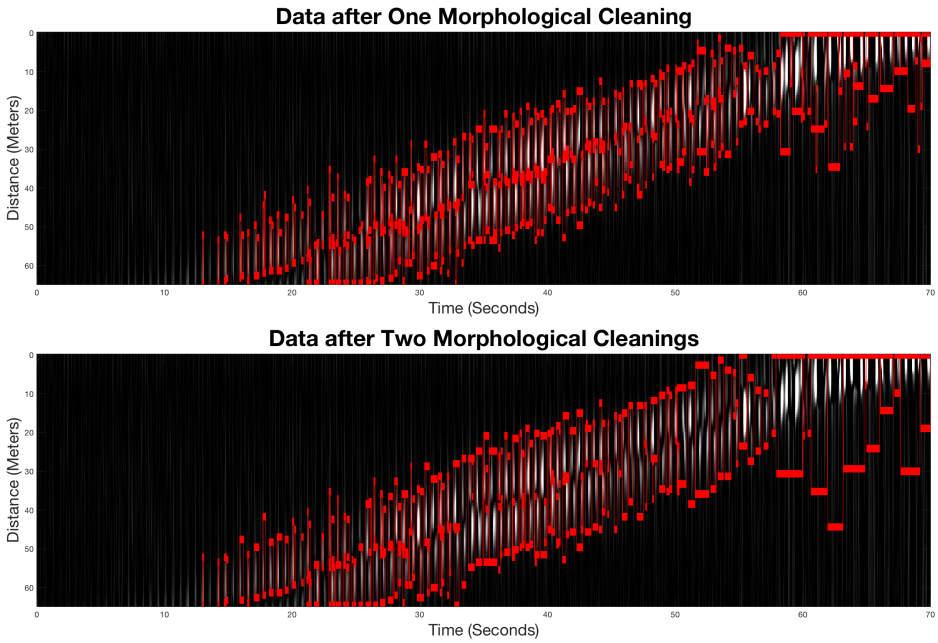

FIG. 3: The vehicle detector trained on the training data applied to the test data after (top) one morphological cleaning and (bottom) two morphological cleanings.

The vehicle detector taught on frames from the training data set does not start detecting events until about 5 seconds after the vehicle begins to show up in the data. The first cleaning shows smaller boxes for events along the vehicle signal toward the beginning and larger bounding boxes at the end. Also, the size of the bounding boxes around events changes sporadically. The second cleaning shows that some of the smaller boxes around events in the signal have disappeared. The boxes still jump significantly in size toward the end of the data. While the signal still appears within the bounding boxes like we want, the large bounding boxes suggests that the vehicle is anywhere within an almost 30m distance. More precision in detection is desirable.

We consider the set of first independent components. Figure 4 shows the resulting first independent components from the segmented data. The yellow vertical lines distinguishes the different frames. With so many segments, it is difficult to determine anything specific about each segment. On close observation, some frames show signs of diagonal movement which would likely be the signal of the vehicle. We will consider the first four segments later in Figure 8.

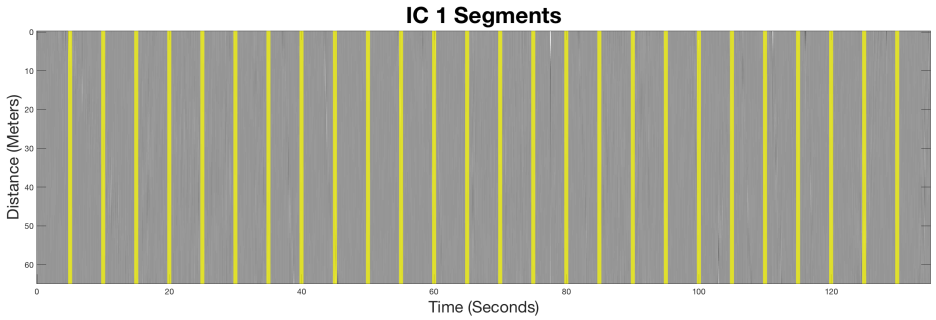

FIG. 4: The 27 segments of the first independent component generated by performing independent component analysis on segments from 20 data sets containing the same 65m segment of fibre over 135 seconds.

Now, we study the results of the applying the vehicle detector trained on the first independent component to the test data. Figure 5 displays the results after the first morphological cleaning on the top and the second morphological cleaning on the bottom. Neither cleaning detects the vehicle's signal as successfully as the test case (seen in Figure 3). The vehicle detector produced bounding boxes with a tighter fit to the vehicle's signal. Most of the bounding boxes around the signal are approximately the same size, which implies that this result gives a better idea of where the vehicle is located along the 65m stretch. The second morphological cleaning shows that the vehicle detector located more of the vehicle's signal after a second cleaning.

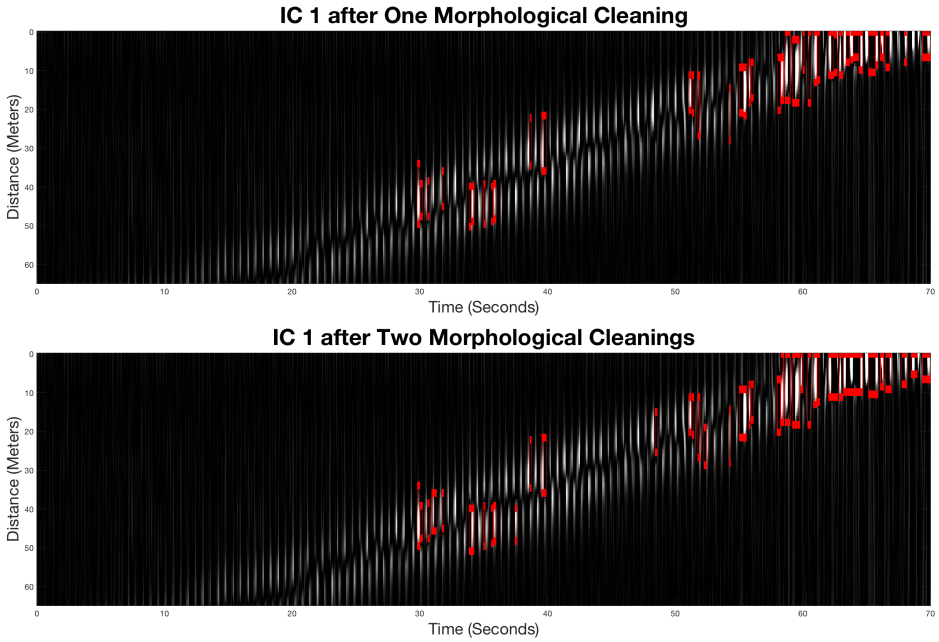

FIG. 5: The vehicle detector trained on the first independent component for each segment of data applied to the test data after (top) one morphological cleaning and (bottom) two morphological cleanings.

Given that we performed the ICA on segments of data, we also consider how the GMM

acts on the second independent components from the segmented data as we cannot guarantee the vehicle signal was only delegated to the first component in each segment. Figure 6 presents the second independent components of each segment where the yellow vertical lines distinguish between the second independent component frames. While difficult to see once again, we can observe some diagonal movement in some of the segments which potentially represents the vehicle moving along the fibre. As with the first independent component, we will consider a few frames of the second component later in the section.

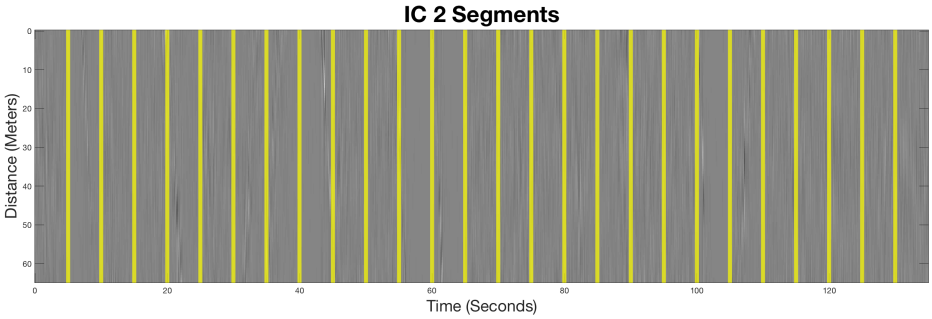

FIG. 6: The 27 segments of the second independent component generated by performing independent component analysis on segments from 20 data sets containing the same 65m segment of fibre over 135 seconds.

Figure 7 displays the results of the vehicle detector trained on the second independent component segments applied to the test data set. The top image shows the results after the first morphological cleaning and the bottom row presents the results after the second cleaning.

The vehicle detector taught using the second independent component segments appears to perform similarly to the vehicle detector trained on the first independent component segments on the test data set. Both sets begin to detect the vehicle's signal at around 30 seconds. The bounding boxes are similarly compact around the signal of the vehicle for this case as well, giving a better estimate of where the vehicle is along the 65m segment of fibre. For the second independent component, it is more evident that the second morphological cleaning improved the results of the vehicle detector as more of the signal is detected between 30 and 45 seconds in that case. The relative success of the vehicle detector for the second independent component on the test data supports our hypothesis that some of the vehicle's signal was delegated to the second component during the independent component analysis of segments from the twenty data sets.

Figure 8 presents the two independent components for the first five segments of data. The first independent components are in the left column and the second independent components are on the right column. We chose two independent components because we wanted to distinguish between the noise and the vehicle signal. The first three frames of the first independent component clearly shows the signal of the vehicle. While the first and third frames of the second independent component are noise, the second frame of the second independent component depicts the signal of the vehicle moving at a higher velocity along the 65m segment of fibre than the vehicle signal in the first component.

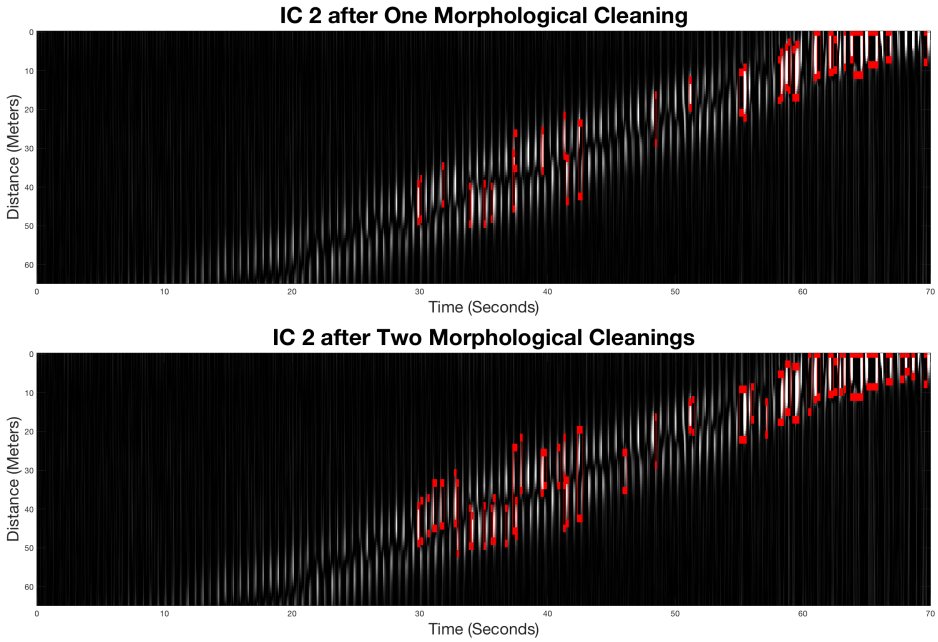

FIG. 7: The vehicle detector trained on the second independent component for each segment of data applied to the test data after (top) one morphological cleaning and (bottom) tow morphological cleanings.

The vehicle signal carries through consecutive frames in the same independent component with relative continuity; however, the vehicle signal is not necessarily delegated to the same independent component once that journey along the fibre is complete as can be seen from the fourth and fifth frame of the second independent component. This supports our claim from earlier that training on the second independent component was appropriate. It should also be noted that while these conditions hold true for the first five frames of the independent components, it is not necessary that these conditions hold true for the remaining twenty-two frames.

Another clear issue with this example is the amount of frames on which the vehicle detector can train the background model. As we discussed in Gaussian Mixture Models section, the vehicle detector needs a number of frames containing only the background before introducing the vehicle signal. Given the performance of all three vehicle detectors in this example, it is essential that we include more training frames, but we also need to ensure that the vehicle detector trains only on background data for the first few frames. While we could employ data sets which gathered data over a longer time period, an easier method finds itself in using the large portion of the data sets to find the independent components and overlapping frames of the data instead of using 5 second disjoint segments. In the next section, we consider the results of independent components from a large segment of multiple data sets instead of multiple segments.

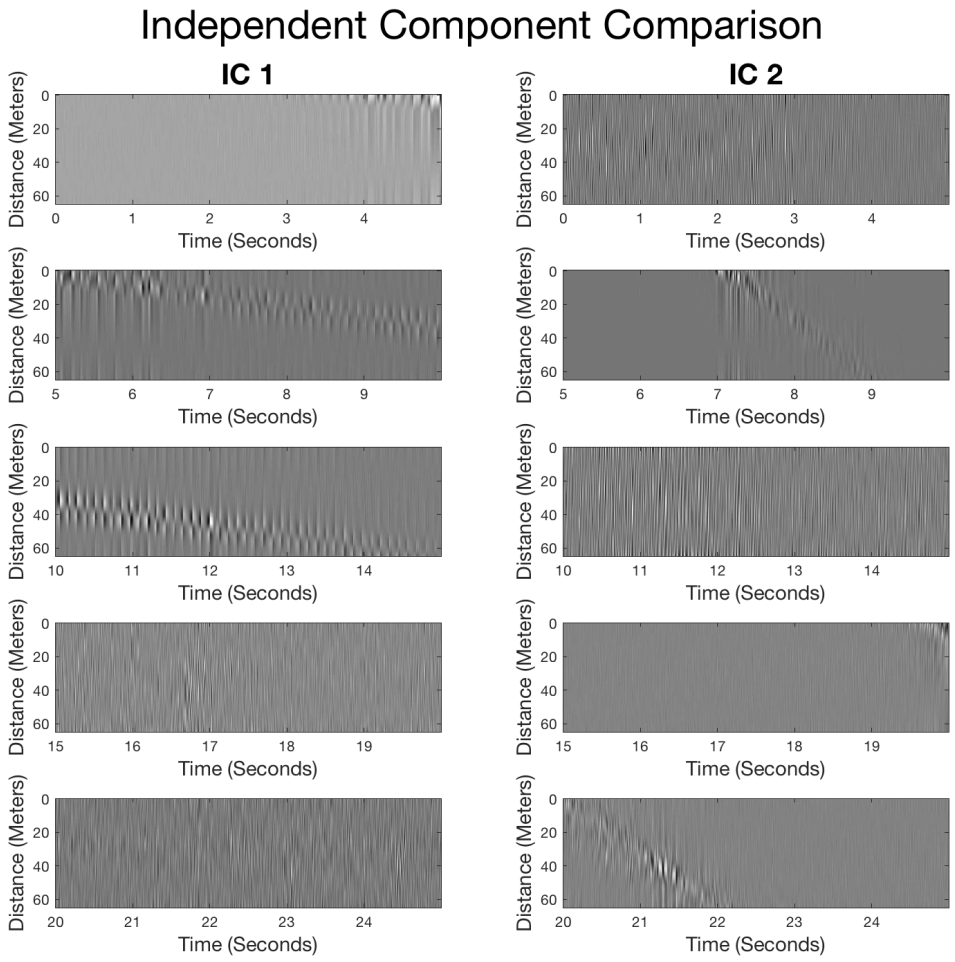

FIG. 8: A comparison of the first five frames of the two independent components. (Left column) The first five frames of the first independent component, covering the time between 0 and 20 seconds. (Right column) The first five frames of the second independent component, covering the time between 0 and 20 seconds.

Case 2: GMM applied to independent components from multiple data sets

We now calculate the first and second independent components by applying an ICA to a portion of all twenty data sets which is the same size as the test data before applying a GMM to distinguish the vehicle from the background. Instead of the entire 135 seconds of data, we only consider 70 seconds of data. Figure 9 shows the two independent components determined by the ICA.

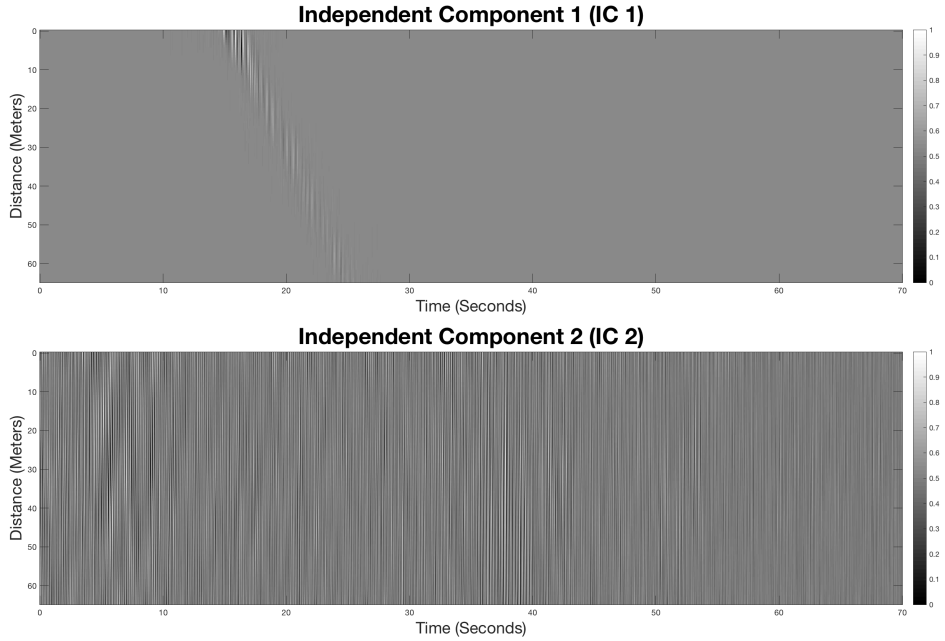

FIG. 9: (Top) The first independent component produced by conducting an independent component analysis on a large segment of twenty data sets generated by the same section of fibre. (Bottom) The second independent component produced by conducting an independent component analysis on a large segment of twenty data sets generated by the same section of fibre.

The first independent component displays the signal of the vehicle driving along the fibre whereas the second independent component contains the noise. We follow a similar pattern to the previous example. We start by considering the results of the vehicle detector trained on the training data set, then the results of the vehicle detector trained on the first independent component, and finally the results of the vehicle detector trained on the second independent component. Afterwards, we set the parameters for each vehicle detector the same as the previous case; however, each detector is taught using 1857 frames, which was approximately $1/3$ of the total frames available for each case.

Figure 10 presents the results of the vehicle detector trained on the training data. In this case, we train the vehicle detector on overlapping frames of the training data and then consider how well it performs on the fourteen segments of the test data as we did in the previous case. Each of the overlapping frames is the same size as the segments from the previous example: 5 seconds by 65 meters.

The top row of Figure 10 shows the results of the vehicle detector after one morpholog-

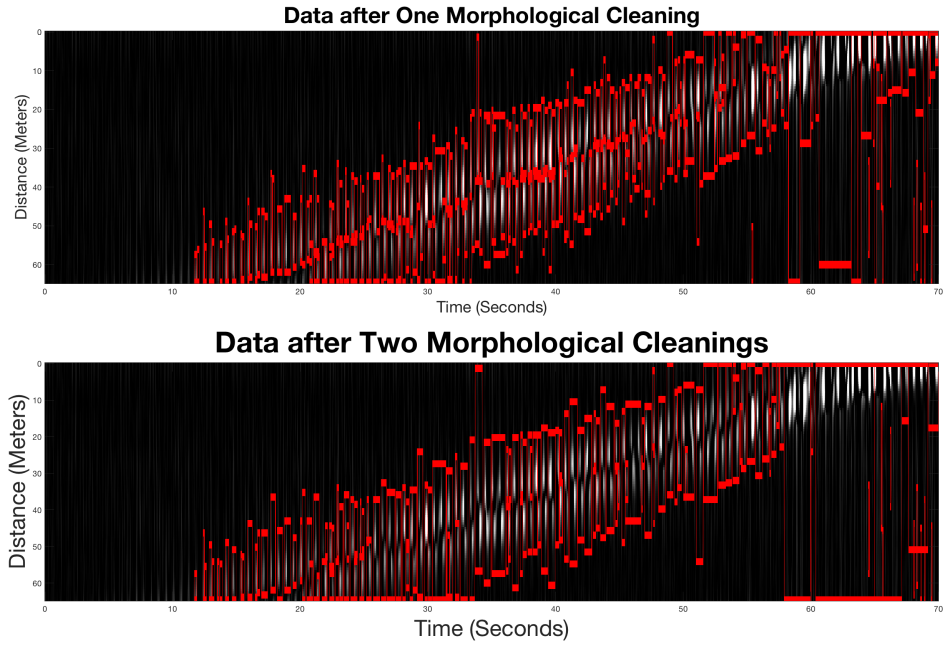

FIG. 10: The vehicle detector trained on the training data applied to the test data after (top) one morphological cleaning and (bottom) two morphological cleanings.

ical cleaning, and the bottom depicts the results after two morphological cleanings. While the vehicle detector does a decent job of detecting the signal from 15 seconds onward for both cleanings, the bounding boxes are quite large especially towards the end. More precision around the signal would be convenient for a better judgment of exactly where the vehicle is along the fibre. The second morphological cleaning reduced the number of boxes by combining them together; however, that is about all that can be determined from observation.

Now, let us consider how the vehicle detector taught on overlapping segments of the first independent component fares for finding the signal of the vehicle; c.f. Figure 11.

This detector performs relatively similarly to the last detector which was trained on the training data. One difference between the two cases is that the bounding boxes are smaller for this vehicle detector, which is especially visible in the final few boxes. Given that the bounding boxes are more concentrated around the vehicle signal suggests that the ICA performed well with regards to separating the signal from the noise. The main difference between the top row and bottom row of Figure 11 is that the bottom row has fewer boxes. In particular, the second morphological cleaning forces smaller boxes to combine.

For the final comparison, let us consider how the vehicle detector which trained on the second independent component performs on the test data; see Figure 12. In the first morphological cleaning, the detector appears to observe something other than the signal of the vehicle between 5 and 10 seconds. After the second morphological cleaning, this bounding box disappears. Perhaps there is more noise than signal in this segment of data, which may be explained by the fact that this detector was trained on the second independent component which contained more noise than the first independent component. This detector

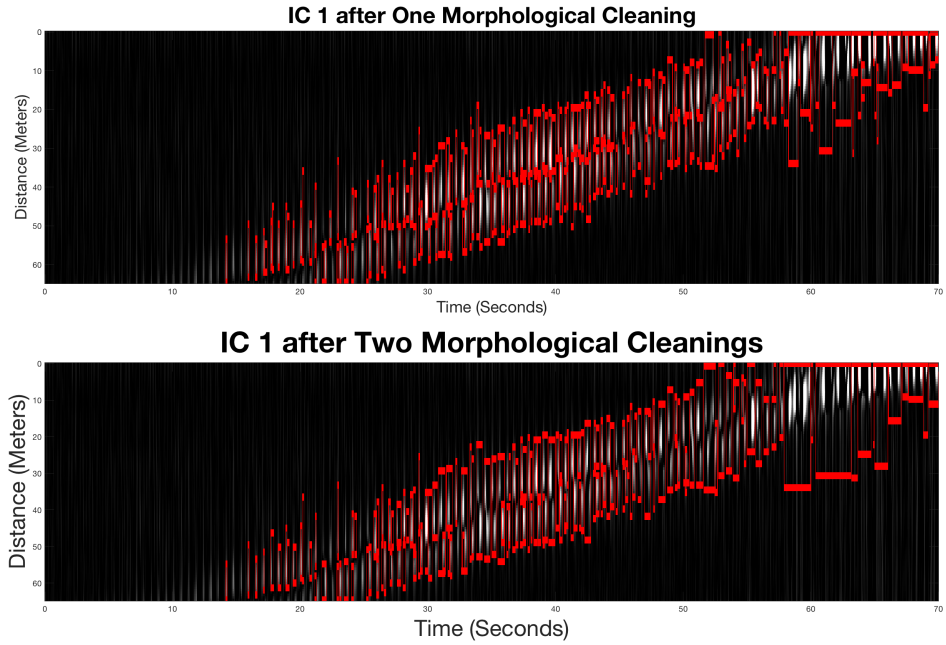

FIG. 11: The vehicle detector trained on a window moving over the first independent component applied to the test data after (top) one morphological cleaning and (bottom) two morphological cleanings.

did not perform as well with regards to detecting the vehicle's signal, unlike the vehicle detector trained on the first independent component. The detector for the first independent component consistently determines the vehicle signal at around 15 seconds whereas the vehicle detector for the second independent component does not consistently detect the vehicle signal until after about 25 seconds.

COMPARISON OF METHODS

Before we conclude, we offer a quantitative comparison of the two cases. In order to provide this comparison, we calculate how many points from the foreground mask are contained within the bounding boxes determined by each vehicle detector and divide the results by the total number of points in the foreground mask for each detector, which we call the overlap percentage. Figure 13 depicts two bar graphs comparing the percentage of points overlapped by each vehicle detector in each case. The case described in Section is on the left and the case from Section is on the right. The height of the bar in the graph provides an idea of how well the method performed on the data, i.e. the taller the bar in the graph, the more points from the vehicle's signal the method detected. The six bars in each graph depict the results of each application of the vehicle detectors to the data:

- (IC1-C1) the first independent component vehicle detector after one morphological cleaning,
- (IC2-C1) the second independent component vehicle detector after one morphological cleaning,
- (D-C1) the training data vehicle detector after one morphological cleaning,

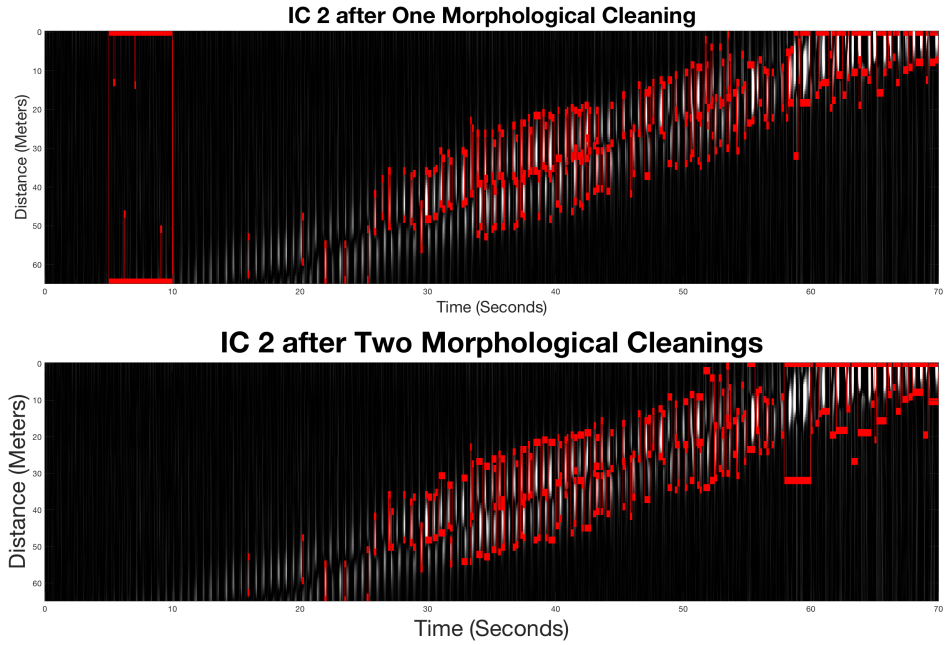

FIG. 12: The vehicle detector trained on a window moving over the second independent component applied to the test data after (top) one morphological cleaning and (bottom) two morphological cleanings.

- (IC1-C2) the first independent component vehicle detector after two morphological cleanings,
- (IC2-C2) the second independent component vehicle detector after two morphological cleanings, and
- (D-C2) the training data vehicle detector after two morphological cleanings.

The training data vehicle detectors for all six cases performed better than any other method. The vehicle detector trained on the segmented training data did not detect the vehicle's signal as well as the vehicle detector trained on overlapping frames of the training data for either morphological cleaning. This follows given that the training set for the first data vehicle detector contained only 10 images compared to the 1857 training frames for the second data detector. This suggests that the number of training frames is important. The vehicle detector for the first independent component performed better than the vehicle detector for the second independent component in all six cases as well. These results support our visual observation that the vehicle's signal was often delegated to the first independent component in many of the cases.

FUTURE WORK

For future work, we plan to explore how well this method works when multiple signals are present in the data; such as, data containing multiple vehicles or a vehicle and pedestrians. We expect that the independent component analysis will be beneficial for separating the different types of signal and will allow the GMM to isolate the specific signal in the

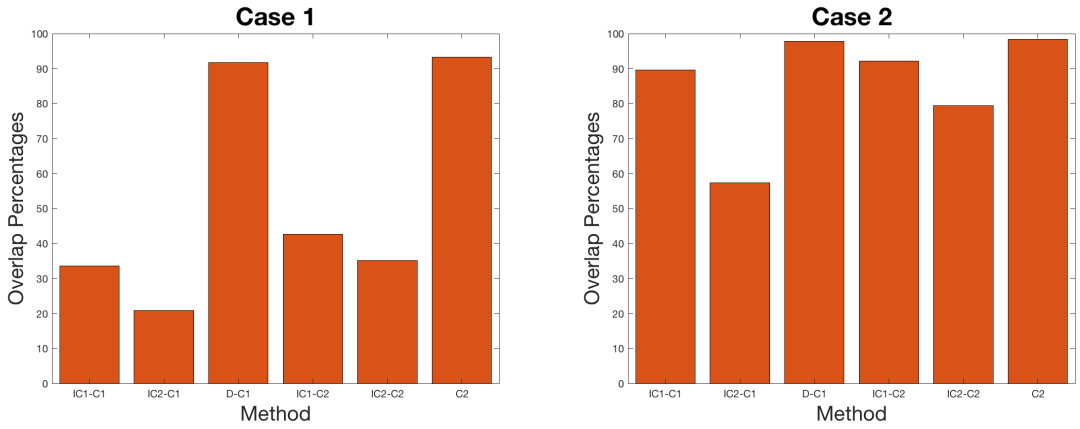

FIG. 13: (Left) The percentage of points in the foreground which are contained by a bounding box for each of the six methods when we train on independent components generated by applying the ICA (left) Case 1 and (right) Case 2. The higher bar indicates how well the method did compared to the others.

data. Without separating the data with multiple signals, the GMM would likely detect each different signal at the same time without distinguishing between them. As such, the independent component analysis would provide the ability to train the GMM on a specific type of signal, which would be beneficial in monitoring projects among other applications.

CONCLUSIONS

In this paper, we employed the ideas from Shamsa and Paydayesh (2019), which used Gaussian mixture models and independent component analysis on microseismic data to highlight events, in applications to data acquired a distributed acoustic sensor. The data contained the signal of a vehicle driving along a road. In each section, we applied independent component analysis to a different ratio of data sets containing the vehicle's signal and noise respectively to compute two independent components. The independent component analysis separated the vehicle's signal from the noise. In some cases, we computed the two independent components from non-overlapping segments, and in other cases, we performed the ICA on a 70 second segment of data. Then, we trained a GMM on a training set and the two independent components.

We found that the vehicle detector trained on the training data statistically performed better than the detectors taught using the independent components. The vehicle detector trained on the independent component containing the vehicle's signal performed the second best when the independent components were generated from 70 second segments of data. Performing ICA on 5 second segments of data provided the worst results in every case.

ACKNOWLEDGMENTS

We thank the sponsors of CREWES for continued support. This work was funded by CREWES industrial sponsors, and NSERC (Natural Science and Engineering Research Council of Canada) through the grant CRDPJ 461179-13, the grant CRDPJ 522863-17,

and the Discovery grant RGPIN-2015-06038 of the third author.

REFERENCES

Gävert, H., Hurri, J., Särelä, J., and Hyvärinen, A., 2005, FastICA 2.5, https://www.mathworks.com/matlabcentral/mlc-downloads/downloads/submissions/49614/versions/2/previews/Thermal%20Pattern%20Separation/FastICA_25/fastica.m/index.html.

Guoshen, Y., 2012, Solving inverse problems with piecewise linear estimators: From Gaussian mixture models to structured sparsity: IEEE Transactions on Image Processing, **21**, No. 5, 2481–2499.

Hartog, A., 2017, An introduction to distributed optical fibre sensors: CRC Press.

Hyvärinen, A., 1999, Fast and robust fixed-point algorithms for independent component analysis: IEEE Transactions on Neural Networks, **10**, No. 3, 626 – 634.

Hyvärinen, A., and Oja, E., 2000, Independent component analysis: Algorithms and applications: Neural Networks, **13**, No. 4-5, 411–430.

KaewTraKulPong, P., and Bowden, R., 2001, An improved adaptive background mixture model for real-time tracking with shadow detection: Springer US.

Shamsa, A., and Paydayesh, M., 2019, Applications of independent component analysis and Gaussian mixture models in micro-seismic signal detection, Geoconvention Partnership, Geoconvention 2019.

Stauffer, C., and Grimson, W. E. L., 1999, Adaptive background mixture models for real-time tracking, *in* 1999 Conference on Computer Vision and Pattern Recognition (CVPR '99), 23-25 June 1999, Ft. Collins, CO, USA, 2246–2252.
URL https://doi.org/10.1109/CVPR.1999.784637

MATLAB, 2019, Detecting cars using Gaussian mixture models, https://www.mathworks.com/help/vision/examples/detecting-cars-using-gaussian-mixture-models.html.