

A diffusion framework for detection of moving vehicles

A. Averbuch, K. Hochman, N. Rabin, A. Schclar and V. Zheludev

School of Computer Science
Tel Aviv University, Tel Aviv 69978, Israel

Abstract

Automatic acoustic-based vehicle detection is a common task in security and surveillance systems. Usually, a recording device is placed in a designated area and a hardware/software system processes the sounds that are intercepted by this recording device to identify vehicles only as they pass by. We propose a novel algorithm for the real-time detection of vehicles based on their recordings. The algorithm uses the wavelet-packet transform in order to extract spatio-temporal characteristic features from the recordings where the underlying assumption is that these features constitute a unique acoustic signature for each of the recordings. The feature extraction procedure is followed by the *Diffusion Maps (DM)* dimensionality reduction algorithm which further reduces the size of the signature. A new recording is classified

by employing the wavelet-packet feature extraction and embedding the result in the reduced-dimension space via the *Geometric Harmonics (GH)* algorithm which is the out-of-sample extension algorithm corresponding to the DM algorithm. The introduced algorithms are generic and can be applied to various signal types for solving different detection and classification problems.

1 Introduction

Sounds emitted by a device can be utilized to identify it via characteristic acoustic features that are inherent in these sounds (assuming that different features are associated with different devices. We refer to these characteristic features as *acoustic signatures*. Motorized vehicles emit sounds whose characteristics are determined by many factors. These factors can be divided into two: factors that depend on the vehicle e.g. engine type, gear system, tires, body and traveling speed and factors that come from the surroundings where the vehicles are traveling e.g. road type (dirt, paved etc.) and condition. A characteristic acoustic signature can be extracted from the sounds of every vehicle. We assume that similar vehicles have similar acoustic signatures while vehicles of different types do not. For example, cars, trucks, airplanes and motorized boats each have a unique acoustic signature. Furthermore, the environment where the vehicles travel influence the sounds the vehicles emit. However, we do not try to characterize this influence and regard it as background noise. This noise makes the identification of vehicles in

various environmental conditions a challenging task. Moreover, background noise that is present in the recording set that is used for the training phase of a detection system poses yet another challenge. Another difficulty is the variability of vehicles' speed and the distance of the vehicles from the microphone. In this paper we wish to separate land vehicles e.g. cars, trucks from non-land vehicles e.g. planes, helicopters via their acoustic signatures.

A common approach for the extraction of acoustic signatures utilizes signal processing techniques e.g. Fourier analysis. An effective tool for this purpose is the wavelet packet transform which we use in the proposed algorithm. We analyze every recording by inspecting the short-term dynamics of the acoustic signal which we represent as vectors of length m .

The proposed algorithm is composed of two steps: (a) an offline training step; and (b) an online detection step. The input to the training step is a set of vehicle and non-vehicle recordings. The acoustic signature of every recording in the set is extracted by applying the wavelet packet transform to it followed by reduction of its dimensionality using the Diffusion Maps algorithm [6]. The signatures are stored for use during the detection step. Although we apply the proposed algorithm for the detection of land vehicles it is generic and can be used for a multitude of detection and identification tasks.

The rest of this paper is organized as follows: in section 2 we survey previous work on vehicle detection. We describe the proposed algorithm in Section 3. Experimental results are given in Section 4. We conclude in Section 4.

2 Related Work

In the following we describe papers that propose different solutions to the problem at hand - many of them are designated for military use.

In [11] speech recognition were utilized in order for the classification of vehicle types. Several speech recognition techniques were compared after the application of the short-time Fourier transform to the recordings. Choe *et al.* [4] utilized the discrete wavelet transform in order to extract the acoustic features. The classification to vehicle types was achieved by a comparison of the feature vectors to a reference database via statistical pattern matching. In [8], a time-varying autoregressive modeling approach was used for the analysis of the signals following the application of the discrete cosine transform. Averbuch *et al.* [2] discriminate between different types of vehicles using features that are derived from the wavelet packets coefficients. The classification of new signals was based on Classification and Regression Trees (CARTs). A similar approach was proposed in [1]. However different techniques were used for the feature extraction and classification. Specifically, the features were extracted using a multiscale local cosine transform that was applied to the frequency domain of the acoustic signal. The classifier was based on the “Parallel Coordinates” [9] methodology. Another paper which makes use of the wavelet packet transform to extract features is [3]. The algorithm distinguishes between vehicles and background. The algorithm incorporates a procedure which searches for a near-optimal footprint. The “eigenfaces method” [13], which was originally used for human face recogni-

tion was utilized by Wu *et al.* [14] in order to classify the acoustic signatures of vehicles. A sliding window was used in order to decompose the signal into a series of short-time blocks. Following the decomposition, the blocks were transformed into the frequency domain. Finally, the dimensionality of the windows in the frequency domain was reduced via Principle Component Analysis (PCA). The classification procedure projected the new signal onto the principle components (after applying a similar decomposition and transform into the frequency domain) that were found during the training.

3 The Proposed Algorithm

The proposed algorithm can be categorized as a supervised learning algorithm i.e. it consists of two steps: training and classification. In the training feature extraction is applied to the training set whose classification is known. This step obtains the acoustic signatures of the recordings and its novelty lies in the feature extraction procedure. The second step classifies new recordings by embedding them into the feature space that was constructed during the training step in order to determine according to trained acoustic signatures whether or not it is a vehicle.

The learning step analyzes a test sample set of recordings $TSS = \{r_i\}_{i=1}^T$ whose classifications are known a-priori where r_i is a recording of length $|r_i|$ and T is the number of recordings in the training set. The recordings do not necessarily have the same size. Each recording r_i is decomposed into overlapping segments $S_i = \{s_j^i\}$. A segment size is denoted by $l = 2^z, z \in$

N. The segments are grouped into a single set $\Sigma = \bigcup_{i=1}^T S_i$. For notational convenience, a single index is used in s_j^i and the output is denoted by $\Sigma = \{s_j\}_{j=1}^{n_s}$ where the total number of segments resulting from the decomposition of all the signals is $n_s \triangleq |\Sigma|$.

Following the decomposition, features are extracted from every segment, feature extraction is applied to segments via the wavelet packets transform. In order to classify a new segment, only a short segment at a time is required (not the entire signal as a batch). Adding to that the high efficiency of the algorithm makes the proposed scheme suitable for real-time applications.

The wavelet packet transform, is an efficient tool that is commonly used for signal processing applications in which both the frequency and the time of the signal are of interest. The proposed extracts features by applying the 6-th order spline wavelet packet, which is described in section A and in more detail in [2, 3]. The features are then classified by using the *Diffusion Maps* framework([6] and Appendix B). Diffusion maps and diffusion distances provide a method for finding meaningful geometric structures in datasets. In most cases, the dataset holds data points in a high-dimensional space \mathbb{R}^n . The diffusion Maps algorithm constructs coordinates that parameterize the dataset and the diffusion distance provides a local preserving metric for this data. A non-linear dimensionality reduction, which reveals global geometric information, is constructed by local overlapping structures. The Diffusion Maps and Geometric Harmonics algorithms are described in Appendix B. For further details on the Diffusion Maps and Geometric Harmonics algorithms, see [6, 7].

3.1 The Learning Step

The first part in the learning step extracts features from each signal. Each signal is divided into segments. Representative features are extracted from each segment. The second part in the learning step reduces the dimensionality of the feature vectors via the application of the Diffusion Maps algorithm. This process is summarized in Algorithm 1:

Algorithm 1 Learning step

1. Application of the wavelet packet transform that uses spline wavelet of order 6 to each segment.
 2. Feature selection: Calculation of the energy distribution of the wavelet coefficients by summing the coefficients in every frequency band in each block.
 3. Every β consecutive segments are averaged in order to reduce the noise.
 4. The learning set is classified via the application of Diffusion Maps.
-

Below is a detailed description of each step in Algorithm 1:

Step 1: The sixth order spline wavelet packet is applied up to a scale $D \in \mathbb{N}$ to each segment $s_j \in \Sigma$. Typically, if $l = 2^{10} = 1024$, then $D = 6$ and if $l = 2^9 = 512$ then $D = 5$. The coefficients are taken from the last scale D . This scale contains $l = 2^z$ coefficients that are arranged into 2^D blocks of length 2^{z-D} . Each block is associated with a certain frequency band. These bands form a near uniform partition of the Nyquist

frequency domain into 2^D parts. The outcome is $U \triangleq \{u_j\}_{j=1}^{n_s}$, where $u_j \in \mathbb{R}^l$. At the end of this step, each segment $s_j \in \Sigma$ is substituted for the set of its spline wavelet coefficients.

Step 2: The acoustic signature of a certain vehicle is constructed by using the distribution of the energy among blocks which consist of wavelet packet coefficients. The energy is calculated by summing the coefficients in every block u_j that was constructed in the previous step. The result of this step is denoted by $E = \{e_j\}_{j=1}^{n_s}$ where $e_j \in \mathbb{R}^{2^D}$. This operation reduces the dimension by a factor of 2^{z-D} .

Step 3: This step is applied to reduce perturbations and noise. The vectors of the set $E = \{e_j\}_{j=1}^{n_s}$ are labeled according to the label of the signal they were originated from. The set E is separated according to the labels and the average of every β consecutive segments, which relate to the same signal, is calculated. The output is denoted by $\tilde{E} = \{\tilde{e}_j\}_{j=1}^{n_a}$ where $n_a = n_s - T \cdot (\beta + 1)$. The number of segments β that are averaged is given as a parameter.

Step 4: The Diffusion Maps algorithm ([6] and Appendix B) is applied to the set \tilde{E} . First, a Gaussian kernel $k(e_i, e_j) = \exp(-\frac{\|e_i - e_j\|^2}{2\varepsilon})$ of size $n_a \times n_a$ is constructed. The kernel is normalized by $P(e_i, e_j) = \frac{k(e_i, e_j)}{d(e_i)}$ where $d(e_i) = \sum_{e_l \in E} k(e_i, e_l)$. Now, P is a Markov transition matrix that corresponds to a random walk on the data with the following eigen-decomposition $p(e_i, e_j) = \sum_{\zeta \geq 0} \lambda_\zeta \psi_\zeta(e_i) \phi_\zeta(e_j)$. The family of diffusion maps $\Psi(e_j) = (\lambda_1 \psi_1(e_j), \lambda_2 \psi_2(e_j), \lambda_3 \psi_3(e_j), \dots)$, embeds the dataset

$\tilde{E} = \{\tilde{e}_j\}_{j=1}^{n_a}$ into an Euclidean space. In these new coordinates, the Euclidean distance between two points is equal to the diffusion distance between the two corresponding high-dimensional points. The diffusion distance measures the connectivity between the two points within the dataset (see Appendix B for the definition of the diffusion distance).

3.2 Classification Step

Denote by $r_t = (r(t - \zeta + 1), r(t - \zeta + 2), \dots, r(t))$ the sequence of ζ signal values that were received up to time t and have to be classified. Each recording r_t is decomposed into β overlapping segments $\{\vartheta_j\}_{j=1}^\beta$ of size l . First, features are extracted from the elements of the set $\{\vartheta_j\}_{j=1}^\beta$. Then, the diffusion maps embedding, which was created in the learning step (section 3.1), is extended to include the new data via the *geometric harmonics* out-of-sample extension algorithm. Finally, each test segment is labeled and assigned to a specific class. These steps are described in algorithm 2.

Algorithm 2 Classification Step

1. Application of the wavelet packet transform (spline wavelet of order 6) to the segments.
 2. Calculation of the energies in the blocks of the wavelet packet coefficients.
 3. Averaging of β consecutive segments.
 4. Extension of the diffusion maps embedding via the geometric harmonics algorithm.
 5. Classification of the new sample according to its nearest neighbors in the diffusion maps coordinates.
-

Step 1: The spline wavelet packet is applied to each segment s_j up to level D . The output is denoted by $\{v_j\}_{j=1}^\beta$, $v_j \in \mathbb{R}^l$.

Step 2: The blocks of energy are calculated similarly to what was done in the learning step. The output is denoted by $\{\epsilon_j\}_{j=1}^\beta$ where $\epsilon_j \in \mathbb{R}^{2^D}$.

Step 3: The β consecutive energy maps $\{\epsilon_j\}_{j=1}^\beta$ are averaged to a single energy map that is denoted by $\tilde{\epsilon}$, $\tilde{\epsilon} \in \mathbb{R}^{2^D}$.

Step 4: The diffusion maps embedding of the training set is extended to the new point $\tilde{\epsilon}$ that was calculated in steps 1-3. The training set \tilde{E} is extended to include the new point $\tilde{\epsilon}$ by applying the geometric

harmonic (Appendix B.2 and [7]) out-of-sample extension algorithm.

The result is given by $\Psi(\tilde{\epsilon}) = (\lambda_1\psi_1(\tilde{\epsilon}), \lambda_2\psi_2(\tilde{\epsilon}), \lambda_3\psi_3(\tilde{\epsilon}), \dots)$,

Step 5: The δ nearest neighbors of the embedded point

$\Psi(\tilde{\epsilon}) = (\lambda_1\psi_1(\tilde{\epsilon}), \lambda_2\psi_2(\tilde{\epsilon}), \lambda_3\psi_3(\tilde{\epsilon}), \dots)$, from the set $\{\Psi\tilde{\epsilon}_j\}_{j=1}^{n_a}$ are found,

where $\delta \geq 1$ is given as a parameter. The classification of r_t is determined according to highest occurring label of its δ nearest neighbors.

4 Experimental Results

The test sample set, (*TSS*) in the learning steps of both algorithms was composed of 42 recordings: 21 of them are recordings of vehicles such as cars, trucks and vans. The remaining part of the test sample set consists of non-vehicle recordings such as airplanes, helicopters, silence, speech, wind, etc. The recordings were sampled at 1000Hz. The following parameters were used for the learning and classification steps of both algorithms: $l = 512$, the overlap between segments was 50%, $\beta = 5$, $D = 6$ and the number of diffusion coordinates was set to be 9. These parameters were determined empirically.

The classification step was tested on recordings that were obtained in various road and wind conditions. For every recording that was used in the classification step we show: (a) the plot of the original signal and (b) the classification probability from the classifier. Probability that is above 0.5 signifies a detection of a vehicle. As we are concerned with the detection of vehicles, an appropriate algorithm should detect vehicles as soon as they

appear in the recording. Furthermore, the number of false positives should be minimal. Once detected, it is of less importance that the probabilities will be above 0.5 throughout the duration the vehicle is heard in the recording. The figures below show that the proposed algorithm exhibits all the required properties.

Figure 1 contains the results of a recording in which a car and a truck pass at times $t = 30 \text{ sec}$ and $t = 50 \text{ sec}$, respectively. The sections between seconds 37 and 43 and between seconds 50 and 60 were part of the *TSS* in the learning step.

Figure 2 shows the results of a recording that contains sounds emitted by a car (around $t = 11 \text{ sec}$) and by another car (around $t = 27 \text{ sec}$). A helicopter is heard starting from $t = 28 \text{ sec}$ until the end of the recording and a third car appears at $t = 70 \text{ sec}$ while the helicopter is still heard. The car at $t = 70 \text{ sec}$ is detected in spite of the helicopter sound in the background.

Figure 4 shows the results for a recording in which a van passes far from the recording device starting at $t = 15$.

Figure 5 contains the results for a recording in which a heavy vehicle is heard twice: first, between $t = 20 \text{ sec}$ and $t = 50 \text{ sec}$ and second, between $t = 90 \text{ sec}$ and $t = 110 \text{ sec}$. During the last 30 seconds of the recording a plane is heard.

The recordings in Figs. 2-5 were not part of the *TSS* that was used in the

training step (Algorithm 1). The classifiers successfully filter out background noise. Both classifiers exhibit very good results and are robust to background noise such as wind.

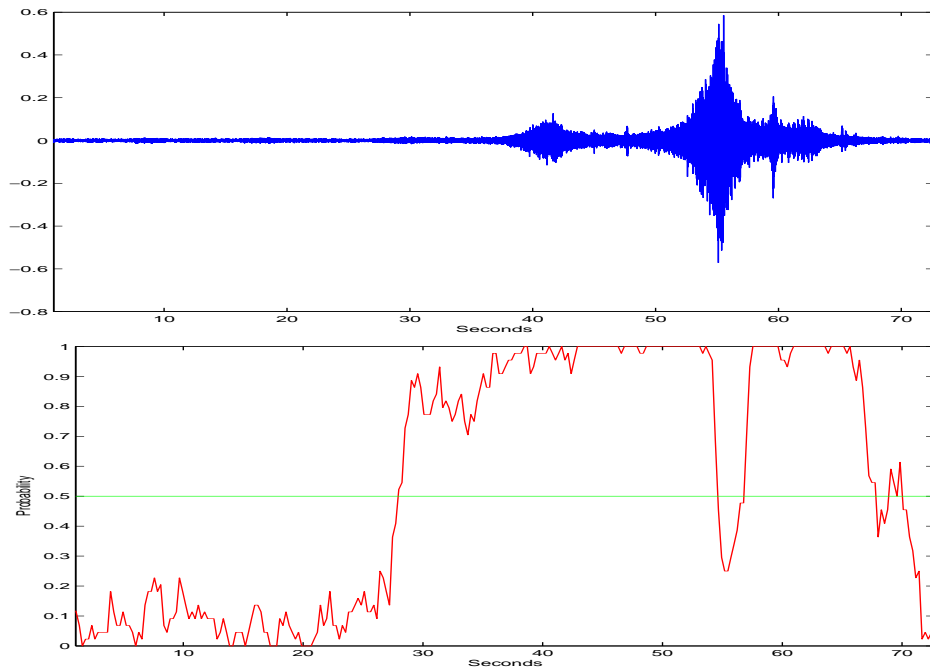


Figure 1: (a) Recordings that contain sounds emitted by a car at $t = 30 \text{ sec}$ and by a truck at $t = 50 \text{ sec}$. The car and truck sections of this recording were part of the TSS in the training step. (b) The vehicle detection probability from the proposed classifier.

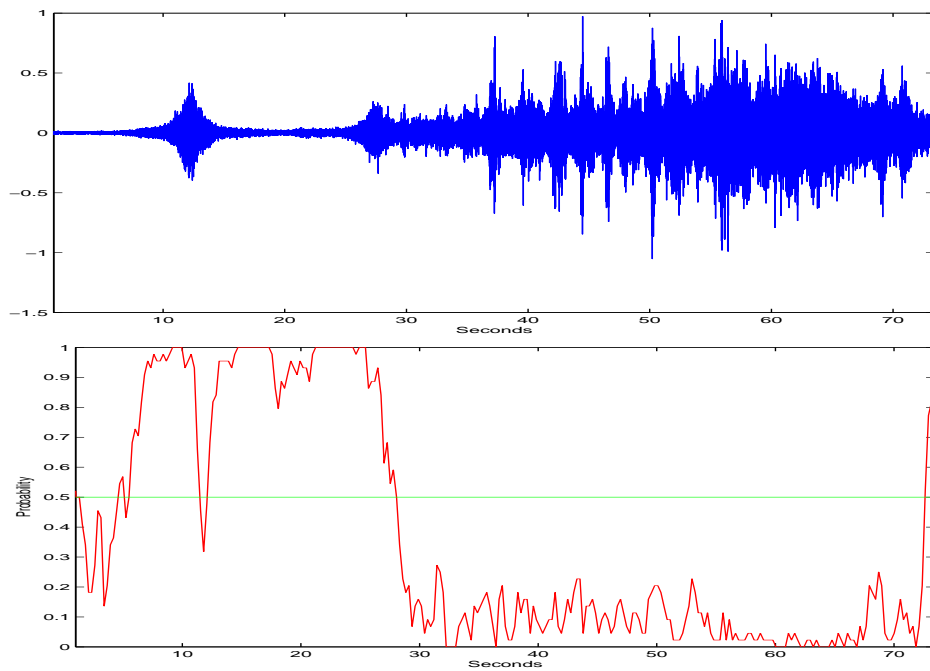
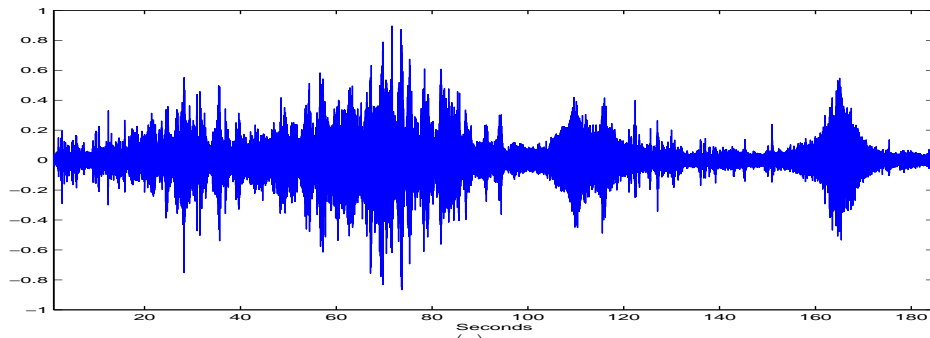
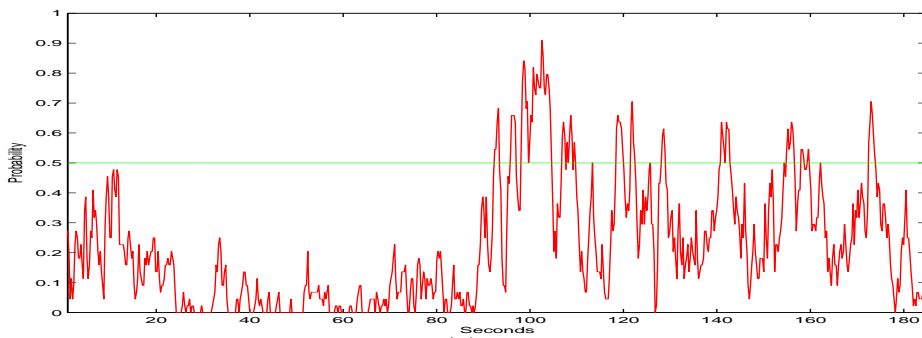


Figure 2: (a) Recordings that contain sounds emitted by a car (around $t = 11 \text{ sec}$) and by another car (around $t = 27 \text{ sec}$). A helicopter is heard starting from $t = 28 \text{ sec}$ until the end of the recording and a third car appears at $t = 70 \text{ sec}$ while the helicopter is still heard. The car at $t = 70 \text{ sec}$ is detected in spite of the helicopter sound in the background. (b) The vehicle detection probability from the proposed classifier.

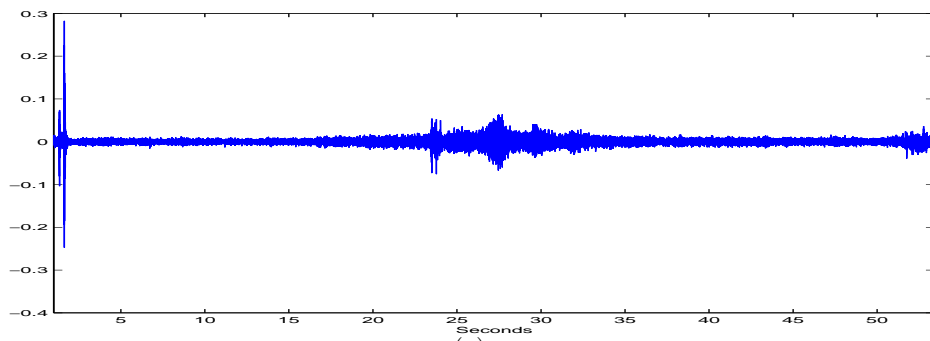


(a)

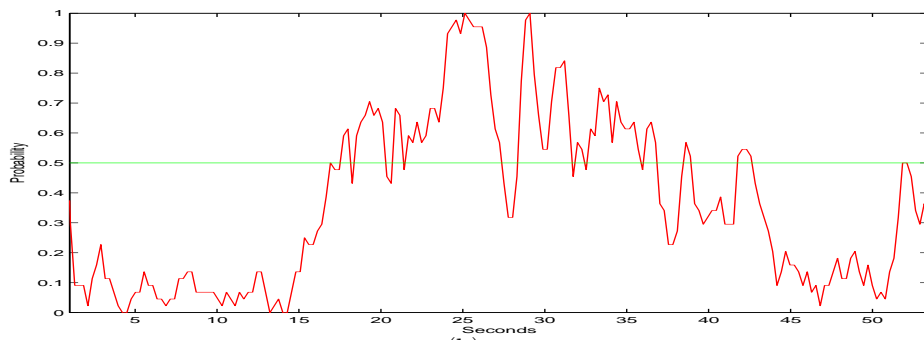


(b)

Figure 3: (a) Recordings in which two helicopters are heard from $t = 15 \text{ sec}$ until $t = 100 \text{ sec}$ and two trucks at high speed pass by around $t = 110 \text{ sec}$ and $t = 160 \text{ sec}$. (b) The vehicle detection probability from the proposed classifier.



(a)



(b)

Figure 4: (a) Recordings in which a van passes far from the recording device starting at $t = 15$. (b) The vehicle detection probability from the proposed classifier.

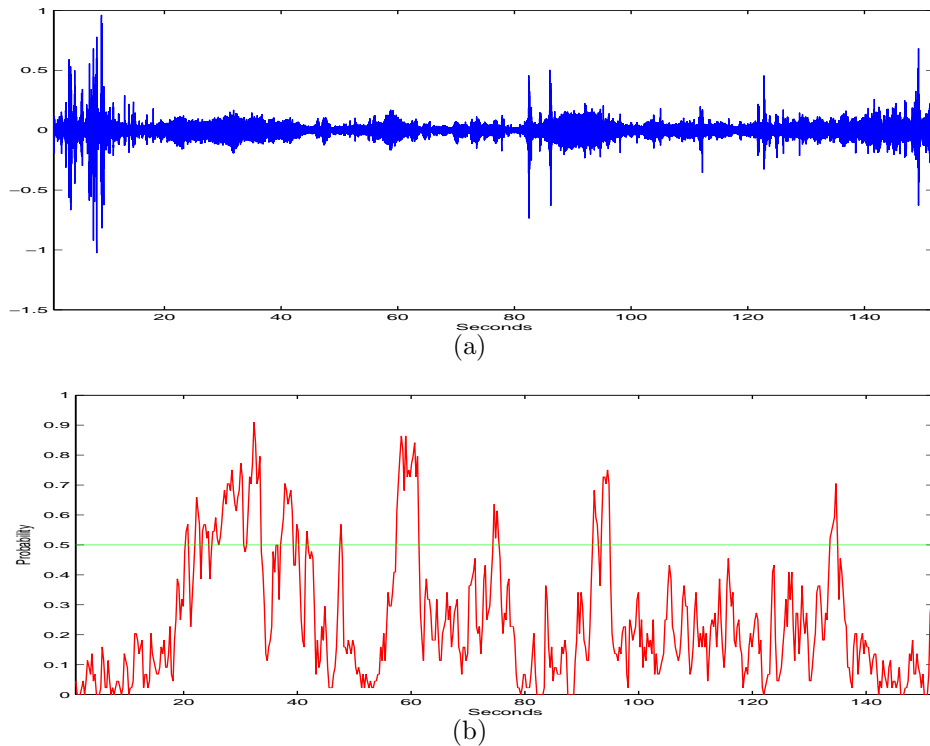


Figure 5: (a) Recordings in which a heavy vehicle is heard twice: first between $t = 20 \text{ sec}$ and $t = 50 \text{ sec}$ and then between $t = 90 \text{ sec}$ and $t = 110 \text{ sec}$. During the last 30 seconds of the recording a plane is heard. (b) The vehicle detection probability from the proposed classifier.

A Wavelet Packet Transforms

The wavelet packet transform is carried out by an iterated application of quadrature mirror filters ($QMFs$) and followed by downsampling. For each such pair of $QMFs$ the high pass filter $G(t)$ and the low pass filter $H(t)$ are applied successively to a finite sampled signal, in order to build a bi-

nary tree of coefficients. For example, the application of the first iteration of the wavelet packet transform on a signal f of length 2^n produces two blocks of size $\frac{n}{2}$, w_0^1 and w_1^1 . The block w_0^1 contains the coefficients for the reconstruction of the low frequency component of the signal. Similarly, the high frequency component can be reconstructed from the block w_1^1 . A second application of the filters G and H on s_0^1 and s_1^1 produces the second level blocks $w_0^2, w_1^2, w_2^2, w_3^2$. All of these blocks are stored in the second level and transformed into eight blocks in the third level, etc. The involved waveforms are well localized in time and frequency domains. Their spectra form a refined partition of the frequency domain (into 2^j parts in scale j). Correspondingly, each block of the wavelet packet transform describes a certain frequency band. Flow of the wavelet packet transform is given by Fig. 6. The partition of the frequency domain corresponds approximately to the location of blocks in the diagram.

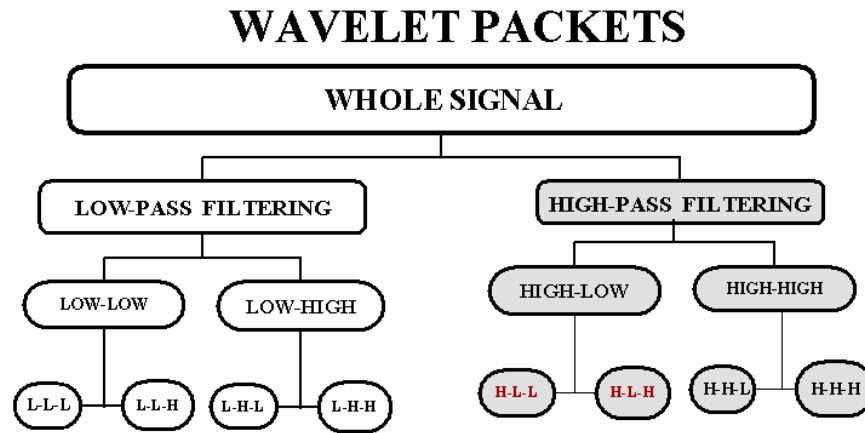


Figure 6: The multiscale decomposition from the application of the wavelet packet transform.

There are many wavelet packet libraries. They differ by their generating filters H and G , the shape of the basic waveforms and their frequency content. In this application, we use the 6-th order spline wavelet packets. Figure 7 shows the wavelet packets after decomposition into three levels generated by wavelet spline of the 6-th order. While the splines do not have a compact support in the time domain, they are localized fairly well in the frequency domain. They produce perfect splitting of the frequency domain.

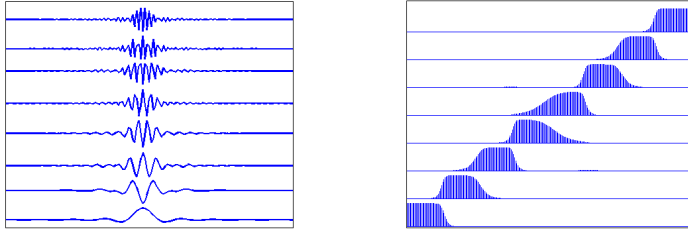


Figure 7: Spline-6 wavelet packets of the third scale (left) and their spectra (right).

B Diffusion Framework

B.1 Diffusion Maps

Let Γ be a set of points in \mathbb{R}^n . The graph $G = (\Gamma, K)$ with a kernel $K \triangleq k(x, y)$, which is a weight function, measures the pairwise similarity between the points is constructed. The weight function is: symmetric: $k(x, y) = k(y, x)$; non-negative: $k(x, y) \geq 0$ for all x and y in Γ ; positive semi-definite: for all real-valued bounded function f defined on Γ $\sum_{x \in \Gamma} \sum_{y \in \Gamma} k(x, y) f(x) f(y) \geq 0$.

The following normalization transforms the kernel into a Markov transition matrix P : $P \triangleq p(x, y) = \frac{k(x, y)}{d(x)}$ where $d(x) = \sum_{y \in \Gamma} k(x, y)$ is the degree of the node x . This normalization is known as the weighted Graph Laplacian normalization [5]. Since P consists of nonnegative real numbers, where each row is summed to 1, the matrix P can be viewed as the Markov transition matrix of a random walk between the points in Γ . The probability to move

from x to y in one time step is given in $P(x, y)$. These probabilities measure the connectivity of the points within the graph.

The transition matrix P is conjugate to a symmetric matrix A given by $a(x, y) = \sqrt{d(x)}p(x, y)\frac{1}{\sqrt{d(y)}}$. Using a matrix notation, $A \triangleq D^{\frac{1}{2}}PD^{-\frac{1}{2}}$, where D is the diagonal matrix with the values $\sum_y k(x, y)$ on its diagonal. The symmetric matrix A has n real eigenvalues $\{\lambda_l\}_{l=0}^{n-1}$ and a set of orthonormal eigenvectors $\{v_l\}$ in \mathbb{R}^n , thus, it has the following spectral decomposition:

$$a(x, y) = \sum_{k \geq 0} \lambda_k v_k(x) v_k(y). \quad (\text{B.1})$$

Since P is conjugate to A , the eigenvalues of both matrices are identical. In addition, if $\{\phi_l\}$ and $\{\psi_l\}$ are the corresponding left and right eigenvectors of P , then we have the following equalities:

$$\phi_l = D^{\frac{1}{2}}v_l, \quad \psi_l = D^{-\frac{1}{2}}v_l. \quad (\text{B.2})$$

From the orthonormality of $\{v_l\}$ and Eq. B.2 it follows that $\{\phi_l\}$ and $\{\psi_l\}$ are biorthonormal i.e. $\langle \phi_m, \psi_l \rangle = \delta_{ml}$. Combing Eqs. B.1 and B.2 together with the biorthogonality of $\{\phi_l\}$ and $\{\psi_l\}$ leads to the following eigendecomposition of the transition matrix P

$$p(x, y) = \sum_{l \geq 0} \lambda_l \psi_l(x) \phi_l(y). \quad (\text{B.3})$$

Because of the fast decay of the spectrum, only a few terms are required to achieve sufficient accuracy in the sum. The family of diffusion maps $\{\Psi(x)\}$ defined by $\Psi(x) = (\lambda_1 \psi_1(x), \lambda_2 \psi_2(x), \lambda_3 \psi_3(x), \dots)$ embeds the dataset into a Euclidean space. This embedding constitutes a new parametrization of the

data in a low-dimensional space. We recall the diffusion distance between two data points x and y as it was defined in [6]:

$$D^2(x, y) = \sum_{z \in \Gamma} \frac{(p(x, z) - p(z, y))^2}{\phi_0(z)}. \quad (\text{B.4})$$

This distance reflects the geometry of the dataset as the value of $\frac{1}{\phi_0(x)}$ depends on the point's density. Two data points are close if there is a large number of paths connecting them. In this case, the diffusion distance is small. Substituting Eq. B.3 in Eq. B.4 together with the biorthogonality property allows to express the diffusion distance using the right eigenvectors of the transition matrix P :

$$D^2(x, y) = \sum_{l \geq 1} \lambda_l^2 (\psi_l(x) - \psi_l(y))^2. \quad (\text{B.5})$$

In these new coordinates, the Euclidean distance between two points in the embedded space represents the distances between the two high dimensional points as defined by a random walk. Moreover, this facilitates the embedding of the original points in a Euclidean space \mathbb{R}^η by:

$$\Xi_t : x_i \rightarrow (\lambda_2^t \psi_2(x_i), \lambda_3^t \psi_3(x_i), \dots, \lambda_\eta^t \psi_\eta(x_i)). \quad (\text{B.6})$$

which also provides coordinates on the set Γ . Essentially, $\eta \ll n$ due to the fast decay of the eigenvalues of P . Furthermore, η depends only on the primary intrinsic variability of the data as captured by the random walk and not on the original dimensionality of the data.

B.2 Geometric Harmonics

Geometric Harmonics (GH) is a method that facilitates the extension of any function $f : \Gamma \rightarrow \mathbb{R}$ to a set of new points that are added to Γ . Specifically, every coordinate of the low-dimensional embedding constitutes such a function. Let Γ be a set of points in \mathbb{R}^n and Ψ be its diffusion map embedding. Let $\bar{\Gamma}$ be a set in \mathbb{R}^n such that $\Gamma \subseteq \bar{\Gamma}$. The GH scheme extends Ψ_t into a new dataset $\bar{\Gamma}$. The GH generalize the *Nyström extension* [12] method which we describe next.

B.2.1 The Nyström extension

The eigenvectors and eigenvalues of a Gaussian kernel on the training set Γ with width ϵ are computed by

$$\lambda_l \varphi_l(x) = \sum_{y \in \Gamma} e^{-\frac{\|x-y\|^2}{2\epsilon}} \varphi_l(y), \quad x \in \Gamma. \quad (\text{B.7})$$

If $\lambda_l \neq 0$, the eigenvectors in Eq. B.7 can be extended to any $x \in \mathbb{R}^n$ by

$$\bar{\varphi}_l(x) = \frac{1}{\lambda_l} \sum_{y \in \Gamma} e^{-\frac{\|x-y\|^2}{2\epsilon}} \varphi_l(y), \quad x \in \mathbb{R}^n. \quad (\text{B.8})$$

This is known as the Nyström extension, which is a common method for extension of functions to out-of-the-training-set points. This is common in online processes in which new samples arrive and we need to extrapolate a function f on the Γ to the new points. The extended distance of $\bar{\varphi}_l$ from the training set is proportional to ϵ . Let f be a function on the training set Γ . In our case, we are interested in extending each of the coordinates of the embedding function $\Psi(x) = (\lambda_1 \psi_1(x), \lambda_2 \psi_2(x), \lambda_3 \psi_3(x), \dots)$.

The eigenfunctions $\{\varphi_l\}$ are the outcome of the spectral decomposition of a symmetric positive matrix, thus, they form an orthonormal basis in \mathbb{R}^n . Consequently, any function f can be written as a linear combination of this basis $f(x) = \sum_l \langle \varphi_l, f \rangle \varphi_l(x)$, $x \in \Gamma$. Using the Nyström extension, as given in Eq. B.8, f can be defined for any point in \mathbb{R}^n by

$$\bar{f}(x) = \sum_l \langle \varphi_l, f \rangle \bar{\varphi}_l(x), \quad x \in \mathbb{R}^n. \quad (\text{B.9})$$

The above extension allows to decompose each diffusion map coordinate ψ_i in the same way using $\psi_i(x) = \sum_l \langle \varphi_l, \psi_i \rangle \varphi_l(x)$, $x \in \Gamma$. In addition, the embedding of a new point $\bar{x} \in \bar{\Gamma} \setminus \Gamma$ can be evaluated in the embedding coordinate system by $\bar{\psi}_i(\bar{x}) = \sum_l \langle \varphi_l, \psi_i \rangle \bar{\varphi}_l(\bar{x})$. Nyström extension, which is given in Eq. B.8, has two major drawbacks:

1. The extension distance is proportional to the value of ϵ used in the kernel. This extension numerically vanishes beyond this distance.
2. The scheme is ill conditioned since $\lambda_l \rightarrow 0$ as $l \rightarrow \infty$.

The second issue can be solved by cutting off the sum in Eq. B.9 keeping the eigenvalues (and the corresponding eigenfunctions) satisfying $\lambda_l \geq \delta \lambda_0$

$$\bar{f}(x) = \sum_{\lambda_l \geq \delta \lambda_0} \langle \varphi_l, f \rangle \bar{\varphi}_l(x), \quad x \in \mathbb{R}^n. \quad (\text{B.10})$$

The result is an extension scheme with a condition number δ . In this new scheme, f and \bar{f} do not coincide on Γ , but they are relatively close. The value of ϵ controls this error. An iterative method for modifying the value

of ϵ with respect to the function to be extended is introduced in [10]. The outline of the algorithm is as follows:

1. Determine an acceptable error for relatively big value of ϵ . Denote them by err and ϵ_0 , respectively;
2. Build a Gaussian kernel using ϵ_0 such that $k_{\epsilon_0}(x, y) = e^{(\|x-y\|^2/2\epsilon_0)}$;
3. Compute the set of eigenvalues for this kernel. Denote this set by $\varphi_l(x)$ and write f as a linear combination of this basis as $f(x) = \sum_l \langle \varphi_l, f \rangle \varphi_l(x)$;

4. Compute the sum

$$r_{err} = \sum_{\lambda_l > \delta \lambda_0} \sqrt{|\langle \varphi_l, f \rangle|^2};$$

5. If $r_{err} < err$ then expand f with this basis as explained in Eq. B.10. Otherwise, reduce the value of ϵ_0 and repeat the process.

The sum, which is computed in step 4 of the algorithm, consists of only large elements. The number of elements in the sum grows with respect to the number of oscillations the function has. This means that ϵ will become smaller with respect to the behavior of f . We are interested in expanding the diffusion maps coordinates that correspond to the highest eigenvalues. These coordinates are usually smooth functions. Thus, the process in steps 1-5 is expected to find a large enough value for ϵ that will allow to extend them to new data points.

References

- [1] A. Averbuch, E. Hulata, and V. Zheludev. Identification of acoustic signatures for vehicles via reduction of dimensionality. *International Journal of Wavelets, Multiresolution and Information Processing*, 2(1):1–22, 2004.
- [2] A. Averbuch, E. Hulata, V. Zheludev, and I. Kozlov. A wavelet packet algorithm for classification and detection of moving vehicles. *Multidimensional Systems and Signal Processing*, 12(1):9–31, 2001.
- [3] A. Averbuch, V. A. Zheludev, N. Rabin, and A. Schclar. Wavelet-based acoustic detection of moving vehicles. *Multidimensional Systems and Signal Processing* - <http://www.springerlink.com/content/32707458802j52t2/>, May 2008.
- [4] H. C. Choe, R. E. Karlsen, T. Meitzler, G. R. Gerhart, and D. Gorsich. Wavelet-based ground vehicle recognition using acoustic signals. *Proc of the SPIE*, 2762:434–445, 1996.
- [5] F. R. K. Chung. *Spectral Graph Theory*. AMS Regional Conference Series in Mathematics, 92, 1997.
- [6] R. R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis: special issue on Diffusion Maps and Wavelets*, 21:5–30, July 2006.

- [7] R. R. Coifman and S. Lafon. Geometric harmonics: a novel tool for multiscale out-of-sample extension of empirical functions. *Applied and Computational Harmonic Analysis: special issue on Diffusion Maps and Wavelets*, 21:31–52, July 2006.
- [8] Kie B. Eom. Analysis of acoustic signatures from moving vehicles using time-varying autoregressive models. *Multidimensional Systems and Signal Processing*, 10:357–378, 1999.
- [9] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1:69–91, 1985.
- [10] S. Lafon, Y. Keller, and R. R. Coifman. Data fusion and multi-cue data matching by diffusion maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1784–1797, 2006.
- [11] M. E. Munich. Bayesian subspace method for acoustic signature recognition of vehicles. *Proc. of the 12th European Signal Processing Conf. EU-SIPCO*, 2004.
- [12] E. J. Nystrom. Uber die praktische auflosung von linearen integralgleichungen mit anwendungen auf randwertaufgaben der potentialtheorie. *Commentationes Physico-Mathematicae*, 4(15):1–52, 1928.
- [13] L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *J. Opt. Soc. Amer. Soc. A*, 4(1), March 1987.

- [14] H. Wu, M. Siegel, and P. Khosla. Vehicle sound signature recognition by frequency vector principal component analysis. *IEEE Transactions on Instrumentation and Measurement*, 48(5):1005–1008, October 1999.