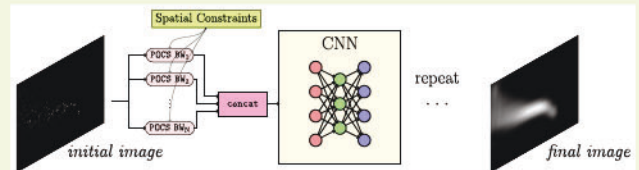


Deep-Learning-Based Gas Leak Source Localization From Sparse Sensor Data

Diaa Badawi^{ID}, Ishaan Bassi^{ID}, Sule Ozev^{ID}, *Member, IEEE*, and A. Enis Cetin^{ID}, *Fellow, IEEE*

Abstract—In this article, we address the problem of estimating the location of gas leak sources using sparse unreliable spatio-temporal chemical sensor data. We pose the task of estimating the underlying gas signal and predicting the source location as an inverse problem. For this purpose, we develop a novel deep-learning projection-based framework. We incorporate traditional projection-onto-convex-sets (POCS) iteration in the structure of the deep model to obtain a regularized solution that conforms to our prior knowledge of the spatio-temporal structure of the gas concentration distribution. We use discrete cosine transform (DCT) layers to model the smooth nature of the gas plume signal. In the DCT domain, we project the feature maps onto a low-pass region, whose boundary is determined during training using the backpropagation algorithm. This operation is equivalent to projecting onto a convex set. Furthermore, these projection operations are embedded in the nonlinear structure of a convolutional neural network. We address two different types of data: methane–propane leak from industrial plants and isopropyl alcohol (isopropanol) vapor leak in an indoor environment. Experimental results are presented. Our results show that we can obtain a smooth estimate of the underlying gas signal while obtaining a good source location prediction with high accuracy.

Index Terms—Deep learning, discrete cosine transform (DCT), inverse problem, multiple-sensor systems, source identification.



I. INTRODUCTION

GAS leak detection is crucial for environmental and security purposes. The rising levels of organic gases such as methane in the atmosphere are very concerning for the environment because of their strong greenhouse effect. This makes developing leak gas detection technology crucial to reducing emissions and combating climate change. On another note, gases such as ammonia (NH₃) are by-products of improvised explosive devices (IEDs) that can wreak havoc on life and property. In light of this, detecting ammonia in these scenarios can immensely help in the early identification of such dangers.

volatile organic compound (VOC) gas detection technologies encompasses methods such as satellite imagery [1], [2], [3], commercially available bolometer-type IR cameras [4], spectrometry [5], and chemical olfactory technology [6], [7]. While these technologies vary vastly based on the application,

chemical sensors are the cheapest and most flexible to use in cyber-physical systems (CPSs) for indoor and outdoor environments. Chemical sensors can detect a wide array of gas analytes such as ammonia, methane, acetone, ethylene, ethanol, and toluene [8]. Nonetheless, one of the most important challenges facing electronic nose technology is the sensor-to-sensor response variations. These variations arise due to aging and environmental factors such as temperature and humidity. This phenomenon is referred to as *sensor drift*. Furthermore, sensor calibration is costly and impractical.

Isopropyl alcohol can be detected by low-cost chemical field effect transistors (ChemFET) [7], [9]. The low cost of this technology and its relative ease to connect to processing units have spurred interest in developing CPSs for dangerous gas detection and source targeting [7]. In [7], a crowd-sourcing-based CPS system for IED detection in large public events is envisioned. In their system, a large number of mobile sensory units are hosted by volunteers among the crowd. These units are attached to the participants' smartphones, and the data is sent to a central server to be analyzed. Given the limited accuracy of the sensors, the CPS employs a two-level feedback logic.

A. Gas Source Localization

While the inherent inaccuracies of sensor measurements are an issue in their own right, determining the source location is challenging even with ideal sensors. Many methods have been

Manuscript received 14 July 2022; accepted 13 August 2022. Date of publication 2 September 2022; date of current version 31 October 2022. This work was supported in part by NSF under Grant 1739396 (UIC) and Grant 1739451 (ASU). The associate editor coordinating the review of this article and approving it for publication was Prof. Xiaofeng Yuan. (Corresponding author: Diaa Badawi.)

Diaa Badawi and A. Enis Cetin are with the Department of Electrical and Computer Engineering, University of Illinois at Chicago, Chicago, IL 60607 USA (e-mail: dbadaw2@uic.edu; aecyy@uic.edu).

Ishaan Bassi and Sule Ozev are with the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ 85287 USA (e-mail: ibassi1@asu.edu; sule.ozev@asu.edu).

Digital Object Identifier 10.1109/JSEN.2022.3202134

suggested to estimate the location of gas source leaks, indoors and outdoor. Traditional approaches rely on expert-knowledge modeling and analyses. One of the most popular models is the Gaussian plume model [10], [11], thanks to its simplicity. Ma *et al.* [10] estimated the model parameters using different optimization algorithms and determine the source location accordingly. Yong *et al.* [12] maximized the likelihood function of a Gaussian plume model in an indoor setting and perform tracking by moving toward the predicted source location. Ma *et al.* [13] adopted a wave dispersion model to locate the gas source based on computational fluid dynamics (CFD). They argued that the downwind distance to the source is linearly correlated with the time it takes gas concentration to stabilize. Other approaches include the use of nonparametric models such as Gaussian processes regressions [14]. In [14], a Bayesian prior of Gaussian processes was assumed and, based upon the density construction, tracking was conducted using simulated annealing to balance between exploration and exploitation. Asenov *et al.* [15] optimized an Eulerian simulator model given with the observed measurements of the sensors and the wind information as the initial conditions. The sensors are mounted on unmanned autonomous vehicles (UAVs). The UAV is then flown to the location that maximizes a likelihood score.

While the previous work explores a wide array of approaches, simulation-based models are complicated and require thorough domain-based fluid dynamics or meteorological knowledge. On the other hand, semianalytical plume-based morphology models assume knowledge of the wind information and assume substantial homogeneity of the wind field. Furthermore, selecting the best model is nontrivial and requires expert knowledge [16].

B. Bandlimited Interpolation Using a Deep Neural Network

The limitations mentioned above motivate us to find a generic framework to address the problem of gas source localization that does not require expert knowledge. For this purpose, we pose the problem as an inverse image-interpolation problem, and, by using the interpolated images, one can predict the source location. Furthermore, we are interested in a data-driven framework to improve the solution to the inverse problem. In this work, we combine traditional image-processing algorithms. In particular, we use the Papoulis–Gerchberg (PG) algorithm [17] and deep convolutional neural networks to achieve this.

Generally speaking, signal and image interpolation problems are ill-posed inverse problems [18]. Without any prior assumptions, there exists an infinite number of solutions. Only a very small set of these solutions is meaningful. The primary assumption we employ is that the gas density signal is *bandlimited* in the Fourier domain. In other words, the signal exhibits low-pass behavior. The bandlimitedness assumption is widely used in many signal-processing applications, especially in the case of sparse input data. In this work, we define bandlimitedness in the discrete-cosine-transform (DCT) domain, instead of the frequency domain, as usual. The main reason is to avoid dealing with complex arithmetic.

In addition, we incorporate a deep neural network in our iterative procedure to introduce data-driven regularization to the ill-posed problem and to overcome the limitations of bandlimited interpolation. We use the backpropagation algorithm to optimize our neural network. After obtaining a gas density image, we predict the source location based on the peak values of the image.

C. Isopropyl Alcohol and Methane Leak Source Identification

In this article, we consider two different scenarios. In the first case, we assume that we have a large number of methane sensors, and we can estimate the gas distribution image (a 2-D function) in a given area. Given that the gas distribution image must be slowly varying and smooth compared to images of rigid objects with sharp edges, we assume a low-pass behavior for the gas distribution as in [19]. In the second case, we have a small number of isopropyl alcohol sensors, and we use the temporal history of the sensor data to predict the location of the isopropyl alcohol gas source. We also assume a low-pass behavior in 1-D sensor measurement data and take advantage of the PG algorithm to estimate the 2-D gas distribution.

The proposed method not only applies to chemical sensors, but also to infrared (IR) and other point sensors.

D. Article Organization

The organization of this article is as follows. In Section II-A, we lay out the mathematical background for bandlimited image interpolation. Furthermore, we explain the deep-learning-based bandlimited interpolation framework we developed. In Section III-A, we describe our isopropyl alcohol data acquisition and the preprocessing of the methane data. In Section IV, we explain our experimental settings and report our results. Finally, we provide our conclusion.

II. DCT-BASED BANDLIMITED INTERPOLATION

Our problem is to estimate the concentration of the chemical gas distribution in a given region from sparsely located sensors. This is essentially an interpolation problem. In the classic image interpolation (completion) problem, one is presented with a subset of image pixel intensities, and the task is to find the missing points from the given point. The interpolated points should conform to a certain assumption made *a priori*. After we interpolate the relative gas distribution, we can identify the maxima of the interpolated distribution and locate the source of the gas leak.

We assume that the gas concentration field is a discrete image $x_g[m, n]$. We approximately know $x_g[m, n]$ at some locations $S = \{(m_i, n_i) \mid i = 1, 2, \dots, L\}$, where L is the number of sensor measurements. We pose the problem as a constrained optimization problem, that is, finding a bandlimited signal whose pixel intensity at the given pixel locations is identical or as close as possible to the intensity of these given pixel values. Interpolating signals via the bandlimitedness assumption has been well studied over the past [20], [21], and it has been successfully applied in many areas such as image super-resolution [22], direction-of-arrival

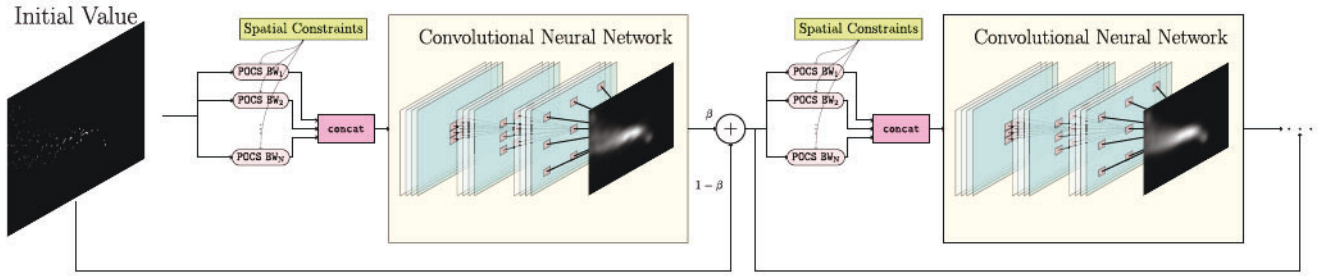


Fig. 1. Block diagram of the proposed data interpolation framework. The POCS procedure is run with different bandwidth parameters and the results are fed to a convolutional neural network. The process is repeated. There are residual connections between each block.

(DoA) estimation [23], and medical imaging [24]. While signals cannot simultaneously be bandlimited and have finite support, a large class of natural finite-support signals has most of their spectral energy concentrated in a finite bandwidth. This means that one can obtain a “good” bandlimited approximation of such signals.

There exist many algorithms to solve bandlimited interpolation problems. One of the classical image interpolation methods is the PG algorithm and its variants [17], [25], [26]. This class of algorithms falls under the more generic framework of alternating projection onto convex sets (POCS).

In the PG algorithm, the problem is posed in a Hilbert space framework, and we are interested in recovering the image $x_g \in \mathbb{R}^{M \times N}$, where M and N are the width and the height of the image. To achieve this, it is assumed that the image x_g is bandlimited in the Fourier domain. This means that the signal lies in the intersection of two convex sets in $\mathbb{R}^{M \times N}$ defined as follows:

$$\mathcal{C}_1 := \{x \in \mathbb{R}^{M \times N} \mid X(\omega_1, \omega_2) = 0 \text{ when } (\omega_1, \omega_2) \notin \text{SUPP}_{\text{BW}}\} \quad (1)$$

where X denotes the discrete-space Fourier transform of an image x , and SUPP_{BW} represents a low-pass region in the Fourier domain, and

$$\mathcal{C}_2 := \{x \in \mathbb{R}^{M \times N} \mid x(m, n) = x_g(m, n) \text{ for } (m, n) \in S\}. \quad (2)$$

The two sets are known to be closed and convex sets in $\mathbb{R}^{N \times M}$ [18]. The PG algorithm is an iterative algorithm since it performs successive orthogonal projections onto $\mathcal{C}_1$ and $\mathcal{C}_2$ to generate a sequence of iterates

$$x^{k+1} = P_1 P_2(x^k), \quad k = \{0, 1, 2, \dots\} \quad (3)$$

where P_1 and P_2 are the orthogonal projection operators onto $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively. If the intersection $\mathcal{C}_1 \cap \mathcal{C}_2$ is nonempty, the iteration converges to a solution $x^* \in \mathcal{C}_1 \cap \mathcal{C}_2$. It is assumed that the intersection is nonempty. Since the sets $\mathcal{C}_1$ and $\mathcal{C}_2$ are closed and convex, the projection operators are firmly nonexpansive operators and so is the composite operator $P_1 P_2$ [18].

Typically, projecting onto the set $\mathcal{C}_1$ is implemented in the Fourier domain. First, the discrete Fourier transform (DFT) of x^k is computed and all the high-frequency components of X^k are forced to zero. Then its inverse DFT is computed. The resulting image may have different values than $x_g[m, n]$ in in

the spatial domain at locations $(m, n) \in S$. The projection onto $\mathcal{C}_2$ imposes the known image values such that $x_{k+1}(m, n) = x_g(m, n), \forall (m, n) \in S$. The process is then repeated until the algorithm converges to a solution or to a fixed point $x^* = P_1 P_2 x^*$. In this article, we use DCT instead of DFT to implement the iterative interpolation algorithm.

Despite its appealing concept, this alternating projection method has its limitations. First of all, the assumption of bandlimitedness does not hold in reality, given that bandlimited signals must have infinite support in the space domain. In reality, the spectrum of natural signals decays very quickly at high frequencies, so one should in principle find an effective “bandwidth” for good interpolation. Nevertheless, signals vary in their spectral energy distribution, and finding an effective bandwidth parameter for each case is nontrivial. This makes the algorithm impractical for automated tasks.

Another concern is that the bandlimited interpolation is optimal in the mean-square sense. This means that interpolating the signal via the aforementioned algorithm is likely to gloss over some fine details that are crucial in some applications, as in locating the gas source in our case.

A. Iterative Projection Onto Convex Sets Using DCT

Let $x_g \in \mathbb{R}^{N \times N}$ be the true gas field image. Since the sensor measurement data is 2-D, we can consider it as an image whose spatial dimensions are the same and equal to N . Extension to three dimensions is straightforward. Let $\mathbf{M}_{\text{NZ}} \in \{0, 1\}^{N \times N}$ be a logical mask such that $\mathbf{M}_{\text{NZ},ij} = 1$ if there is a nonzero measurement at location (i, j) and zero otherwise. Let $\mathbf{M}_Z \in \{0, 1\}^{N \times N}$ be a logical mask such that $\mathbf{M}_{Z,ij} = 1$ if the reading at location (i, j) is equal to 0 and 1 otherwise. Let $\circ$ denote the Hadamard (elementwise) matrix product. Given the sparse input map $x_s := x_g \circ \mathbf{M}_{\text{NZ}}$, our objective is to recover the missing points $x_g \circ (1 - \mathbf{M}_{\text{NZ}} - \mathbf{M}_Z)$. In other words, we want to find an estimate $\hat{x} \in \mathbb{R}^{N \times N}$, such that $\hat{x} \circ \mathbf{M}_{\text{NZ}} = x_s = x_g \circ \mathbf{M}_{\text{NZ}}$ and $\hat{x} \circ \mathbf{M}_Z = 0$. The signal $\hat{x}$ must also be bandlimited. Finally, $\hat{x}$ must be nonnegative since the true gas field signal is nonnegative. The nonnegativity constraint also corresponds to a half-space which is also a convex set.

The constraints $\hat{x} \in \mathcal{C}_2$ is expressed as follows:

$$[\hat{X}]_{ij} = [\mathcal{T}_{\text{DCT}} \hat{x}]_{ij} = 0 \quad \forall (i, j) \notin \text{SUPP}_{\text{BW}} \quad (4)$$

where $\mathcal{T} : \mathbb{R}^{N \times N} \mapsto \mathbb{R}^{N \times N}$ is the 2-D type-II DCT, and SUPP_{BW} is a discrete set $\subset \{0, 1, \dots, N-1\} \times \{0, 1, \dots, N-1\}$.

We defined the bandwidth support (region) as

$$SUPP_{BW} := \{(i, j) | i + j \leq BW\} \quad (5)$$

for some integer number BW. This corresponds to a triangular area in the DCT domain. Given the DCT $\hat{X}$ of an image, the projection onto the DCT-bandlimited subspace is given by

$$[P_{DCT}\hat{X}]_{ij} := \begin{cases} [\hat{X}]_{ij}, & i + j \leq BW \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

The projection onto the $\mathcal{C}_2$ is simply done by equating the interpolated image with the given spatial constraints $\hat{x} \circ M_{NZ} = x_s = M_{NZ} \circ x_g$ and $\hat{x} \circ M_Z = 0$. The alternating projection procedure is summarized as pseudocode in Algorithm 1.

Algorithm 1 Pseudocode for the DCT-Based Iterative POCS. $\hat{x}_0$ Is the Initial Estimate, x_s Is the Given Input Measurements, M_{NZ} and M_Z Are Both the Nonzero and the Zero Measurements Masks. $\hat{x}_i$ Is the Interpolated Image at Step i . I Is the Total Number of Iterations. Lower-Case x and y Are in the Space Domain, While Uppercase X and Y Are in the DCT Domain. $\alpha \in \mathbb{R}$ Is an Acceleration Factor Given in (7), and $(\cdot)_+$ Zeros Out Negative Values

Input: $\hat{x}_0, x_s, M_{NZ}, M_Z, I, BW$
1 create projection mask P_{DCT} with parameter BW
2 for $i = 1$ to I do
3 $X = T_{DCT}\hat{x}_i$ (take DCT)
4 $Y = P_{DCT}X$ (Bandlimiting)
5 $y = \alpha T_{DCT}^{-1}Y$ (inverse DCT)
6 $y = M_{NZ} \circ x_g + (1 - M_{NZ}) \circ y$
7 $y = (1 - M_Z) \circ y$
8 $\hat{x}_{i+1} = (y)_+$
9 end
10 return $\hat{x}^I$

The acceleration factor α in Algorithm 1 is a normalizing scalar that can be chosen to accelerate the convergence. In this work, we set α to

$$\alpha = \frac{\langle M_{NZ} \circ (T^{-1}Y), x_s \rangle}{\|M_{NZ} \circ (T^{-1}Y)\|_2^2 + \epsilon} \quad (7)$$

which minimizes the mean-square error between the intermediate values of y at the given locations and the given constraints $x_s = M_{NZ} \circ x_g$, and ϵ is a small number to ensure numerical stability. This accelerated version of the algorithm was first proposed in [27] and is proved to converge to a solution, provided one exists, that is, the intersection of the sets is nonempty.

B. Deep Regularizer-Based Iterative Completion

It is well known that the bandlimited interpolation is an ill-posed problem [18], [20]. This means that, even with the bandlimiting assumption, the solution can still be far from the desired one. In general, one can add regularization functions to the bandlimiting interpolation problem and solve the following

optimization problem:

$$\begin{aligned} \min & f(x) \\ \text{s.t. } & x \in \mathcal{C}_1 \\ & x \in \mathcal{C}_2 \\ & x \geq 0. \end{aligned} \quad (8)$$

Youla and Webb [18] used the energy of the image, that is, the square of the ℓ_2 norm as $f(x)$ to regularize the image construction. In general, for a subdifferentiable regularizer function, one can solve the problem posed in (8) using the following iteration:

$$\begin{aligned} y &= \text{POCS}(x^i) \\ x^{i+1} &= \text{Prox}_{\lambda f}(y) \end{aligned} \quad (9)$$

where POCS refers to the procedure described in Algorithm 1 and $\text{Prox}_{\lambda f}(y) := \arg\min_z \{(1/2)\|y - z\|_2^2 + \lambda f(z)\}$ is the proximal operator of the regularizing function $f(\cdot)$ with hyperparameter $\lambda > 0$. If the function $f(\cdot)$ is differentiable, the proximal operator reduces a regular gradient-descent update step.

Nevertheless, one of the main difficulties involving the previous procedure is to find an appropriate regularizer function $f(x)$, whose proximal operator can be calculated easily. Instead, we use deep learning to learn a data-driven regularization procedure. This practice has been applied in many approaches, such as learned iterative shrinkage algorithm (ITSA) and deep-based landweber iterations [28], [29], [30]. Instead of directly learning a regularizing function $f(x)$, one can replace the second step in the procedure in (9) with a function $\text{DNN}(y)$, where DNN stands for a deep neural network. The procedure now becomes

$$\begin{aligned} y &= \text{POCS}(x^i) \\ x^{i+1} &= \text{DNN}(y) \end{aligned} \quad (10)$$

where DNN is a deep neural network that we train using past data for the task. The forward model $\text{DNN}(\cdot)$ can be interpreted as applying a data-driven regularization step.

C. Interpolation With Unknown Bandwidth

Another important reason for incorporating deep learning in the POCS approach is to overcome the fact that we do not know the bandwidth parameter suitable for each case, as mentioned before. For this purpose, we propose solving the problem with different preset bandwidth parameters in parallel and let the deep neural network in the procedure given in (10) fuse the different solutions and decide on which one fits the observed data.

To achieve this, we run the DCT-based POCS algorithm for different bandwidth values in parallel. Let the set of bandwidth parameters be $\mathcal{BW}$, where each parameter $BW \in \mathcal{BW}$ correspond to a different bandwidth parameter as in (5). Let $B := |\mathcal{BW}|$ be the number of bandwidth parameters. After running POCS for all cases in $\mathcal{BW}$ in parallel, we concatenate the resulting images and feed them to our DNN: $\mathbb{R}^{N \times N \times B} \mapsto \mathbb{R}^{N \times N}$. It is worth mentioning that the neural network will output a single image. This allows the

neural network to construct the output image from different choices of bandwidth. The resulting image is then fed back to the POCS algorithm and the process is repeated L times.

The algorithm is summarized in Algorithm 2 and a visual illustration of the framework is shown in Fig. 1. Note that we use a skip-connection (dampening factor) $\beta \in [0, 1]$ when combining the results of the previous POCS iterations and the previous output of the neural network, as indicated in line 5 in Algorithm 2. The variable y_j^i corresponds to the intermediate output of the residual operation. This approach improves the flow of gradient in the backpropagation algorithm and, thus, accelerates the training process. The inner loop in Algorithm 2 (lines 4–7) is run in parallel using vectorization semantics.

It is worth mentioning that, by using a fast implementation of DCT, the POCS procedure runs at a comparable speed to a relatively small CNN that implements separable convolution. Performing POCS for K iterations for all bandwidths in parallel accounts to $O(N \times N \times \log N \times B \times K)$ complexity. For a separable convolutional layer with an input of size $N \times N \times B'$ and an output of size $N \times N \times B'$, where B' is the number of channels, calculating the channelwise convolution is $O(m \times m \times N \times N \times B')$ for kernels of size $m \times m$, and calculating the pointwise convolution is $O(N \times N \times B' \times B')$. Usually, $B' > m \times m$. Therefore, for a network of K layers, the order of the number of operations is $O(N \times N \times B' \times B' \times K)$, compared $O(N \times N \times \log N \times B \times K)$. In our case, B is 4, which is much smaller than a typical choice of the number of channels B' in a neural network, as the number of channels can go up to 2048 in very deep models. Given that $\log N < B'$, running the POCS procedure for K steps for B different bandwidth parameters is on par with running a very small convolutional network of K layers with small channel (depth) size ($B' = B$).

Algorithm 2 Pseudocode for the Deep Regularizer DCT-POCS Procedure. The Algorithm Takes the Sparse Input x_g and the Two Logical Masks M_{NZ} and M_Z , the Number of Unrolling Steps Is L , the Bandwidth Parameters Set BW , and the Number of Unrolling Steps for the POCS Algorithm K . The Algorithm Returns the Output $\hat{x}^L$ that Represents the Completed (Interpolated) Image Data. The Subroutine $\text{POCS}(x_j^i, x_s, M_{NZ}, M_Z, BW_j, K)$ Implements Algorithm 1 With Initial Condition y_j^i and Iterates for K Steps

```

Input:  $x_s, M_{NZ}, M_Z, L$ 
1  $\hat{x}^0 \leftarrow x_s$ 
2  $y_j^0 \leftarrow x_s$  ( $j=1$  to  $B$ )
3 for  $i=1$  to  $L$  do
4   for  $j=1$  to  $B$  do
5      $y_j^i \leftarrow \beta y_j^{i-1} + (1 - \beta)\hat{x}^{i-1}$ 
6      $z_j^i \leftarrow \text{POCS}(y_j^i, x_s, M_{NZ}, M_Z, BW_j, K)$ 
7   end
8    $Z^i \leftarrow \text{Concat}(z_1^i, z_2^i, \dots, z_B^i)$ 
9    $\hat{x}^i \leftarrow \text{DNN}(Z^i)$ 
10 end
11 return  $\hat{x}^L$ 

```

TABLE I

ARCHITECTURE OF THE DEEP REGULARIZED NETWORK USED IN THE LARGE-SCALE METHANE DATA EXPERIMENT. THE FILTER SIZE IS 3×3 IN THE LAYERS EXCEPT THE FIRST AND THE LAST (CONV 1 AND CONV 7), WHERE IT IS SET TO 5×5

Layer	Num Channels	Dilation Rate	Activation
Input ($128 \times 128 \times 4$)			
Conv 1	64	1	Leaky Relu (0.2)
Conv 2	64	2	Leaky Relu (0.2)
Conv 3	64	4	Leaky Relu (0.2)
Conv 4	64	8	Leaky Relu (0.2)
Conv 5	64	4	Leaky Relu (0.2)
Conv 6	64	2	Leaky Relu (0.2)
Conv 7	1	1	Relu
Output ($128 \times 128 \times 1$)			

TABLE II

ARCHITECTURE OF THE DEEP REGULARIZED NETWORK USED IN THE SMALL-SCALE ISOPROPANOL DATA EXPERIMENT. THE FILTER SIZE IS 3×3

Layer	Num Channels	Dilation Rate	Activation
Input ($32 \times 32 \times 8$)			
Conv 1	16	1	Leaky Relu (0.2)
Conv 2	16	2	Leaky Relu (0.2)
Conv 3	16	4	Leaky Relu (0.2)
Conv 4	16	2	Leaky Relu (0.2)
Conv 4	1	1	Relu
Output ($32 \times 32 \times 4$)			

To train the deep model, we unroll a few layers of the procedure and optimize the neural network to minimize the loss criterion. We use two loss criteria for the large-scale methane–propane and the small-scale indoor isopropyl alcohol interpolation problems as explained in Section IV. We also use two different network architectures. The first architecture is for the large-scale 128×128 input images correspond to methane–propane gas field signals, and another $32 \times 32 \times 8$ temporal low-resolution images correspond to the indoor isopropyl alcohol leak data that we collected using three commercially available tin oxide (SnO_2) MQ317 chemical sensors that are sensitive to isopropyl alcohol [31]. Details on the datasets are provided in Section III-A. The architecture of the large-scale neural network for the methane data experiment is given in Table I. The architecture of the small-scale neural network for the isopropanol data experiment is given in Table II. Note that in this case the neural network $\text{DNN} : \mathbb{R}^{32 \times 32 \times (T \times |BW|)} \mapsto \mathbb{R}^{32 \times 32 \times T}$, where T is the temporal dimension and K is the number of bandwidth choices.

III. DATA ACQUISITION AND PROCESSING

A. Leak Detection Using Chemical Gas Sensors

We collected sensor measurements using three tin oxide (SnO_2) MQ137 sensors [31]. Data collection was done using an Arduino, as the sensors have an analog output wired connection and were used for reliability. For the experiments of source localization, a grid of $18 \times 18 \text{ in}^2$ was created. As the sensors are resistive chemical sensors, they need to

be preheated for a certain period of time before testing. This is done so that the sensors form a desired active oxide layer than can detect any gas present in the air. In this case, before starting the experiments, we heated the sensors for up to 48 h. These sensors are not calibrated, so to do sensor-to-sensor calibration, first, a controlled experiment was done. In these controlled experiments, we placed the three sensors close to one another so that they all read the same underlying gas concentration signal. Ideally, the relation between the sensor measurement and the actual concentration can be approximated by the following formula:

$$s(t) = Ac(t) + b \quad (11)$$

where $s(t)$ is the sensor measurement (in volt) and $c(t)$ is the gas concentration (in ppm). b is the dc offset, and A is a gain factor. Each sensor will have its own parameters A and b . We first estimated the dc offset using the measurement values when no gas signal is present. We then calculated the relative gains of each sensor with respect to a reference sensor. Let sensor s_1 be our reference sensor, and the relative gain is given by

$$\frac{A_i}{A_1} = \frac{1}{T} \sum_{t=1}^T \frac{s_i(t) - b_i}{s_1(t) - b_1}. \quad (12)$$

We then calibrated the sensor measurements using the following formula:

$$\hat{c}_i(t) = \frac{A_1}{A_i} (s_i(t) - b_i). \quad (13)$$

One issue with this calibration approach is that the actual sensor response can deviate significantly from the ideal model expressed in (11). In order to get reliable and robust estimates for the relative gains, we manually selected segments for calibration in which the sensor responses are linearly correlated the most.

According to our experiments, we found that the dc offsets b_1 , b_2 , and b_3 are equal to 0.7 V. The relative gains are $(A_2/A_1) = 1.43$ and $(A_3/A_1) = 1.61$.

We then applied the calibration formulas using these numbers and fed the input to our deep model to determine the source location as explained in detail in Section IV-B.

The experiment was conducted in the following order.

Step 1: The source was placed at the desired location.

Step 2: The three sensors were placed at different grid points as shown in Fig. 2.

Step 3: Once the source and the sensors were placed at their respective locations, the lid of the source was removed, exposing the air.

Step 4: The data was collected from each sensor using the Arduino which transferred the data to a computer.

Step 5: After one set of experiments was completed, the lid of the source was closed and the room air was vented out. We waited until all the sensors returned back to their baseline response, indicating that there is no more gas in the air.

Step 6: The procedure was repeated six times with different source and sensor positions on the defined grid and corresponding sensor data was collected each time. This data was then used in our source localization framework.

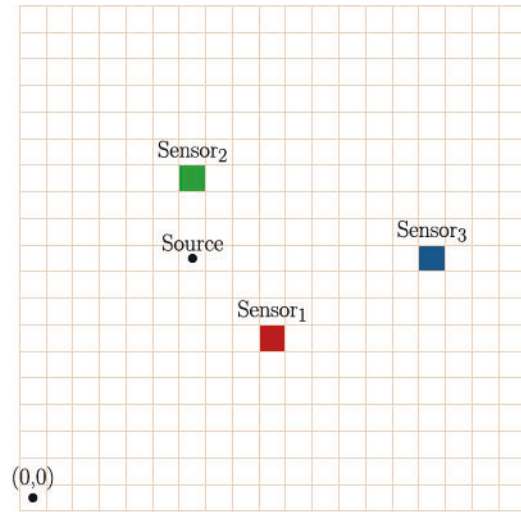


Fig. 2. Illustration of the isopropyl alcohol 18×18 in² grid with the source stationed at $(x = 6, y = 9)$ and the three sensors located at $(x = 9, y = 6)$, $(x = 6, y = 12)$, and $(x = 15, y = 9)$. The corresponding time series are shown in Fig. 3.

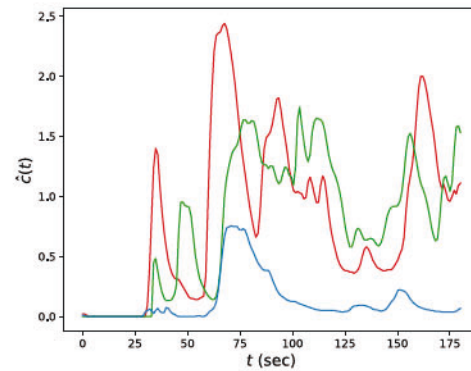


Fig. 3. Corresponding calibrated time-series measurements of each sensor in the experiment shown in Fig. 2.

Fig. 3 shows how the data from the sensors look for one set of experiments.

B. Methane Data Preprocessing

In this section, we simulate an IR sensor system using IR video cameras. We used the dataset collected in [11]. The dataset consists of high-resolution thermal-IR imaging video recordings of gas leaks from industrial plants. Given that methane has high absorbance in the long-wave IR (LWIR) spectrum, methane leakage can be detected via IR imaging [32]. This means that pixel intensities and their temporal behavior are directly indicative of the presence of methane signals at their respective locations.

In order to pose the problem as a source identification from sparsely located IR sensor data, we first preprocess the videos by removing the background from the scene. This was done by estimating the optical flow using the Kanade–Lucas algorithm [33]. We segmented the foreground from the background based on whether the magnitude of the optical flow field exceeds a certain threshold. Given that the camera is stationary

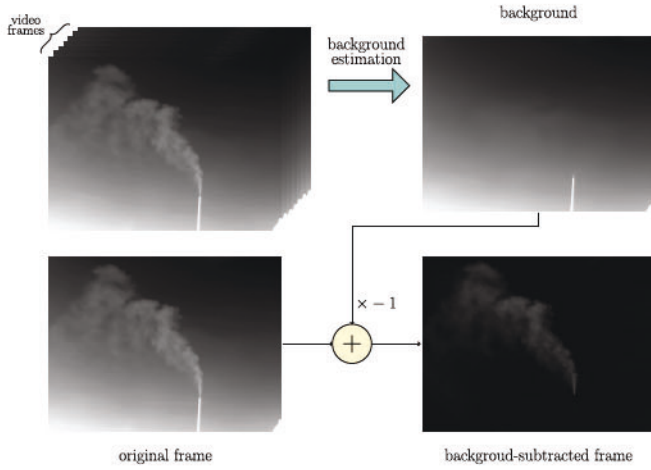


Fig. 4. Illustration of the background subtraction process.

in these videos, we estimated the background by taking the average over the different background estimates. We then completed the missing parts using the PG algorithm and manually smoothed the boundaries. Fig. 4 shows an example of an original video frame and the processed result. Sparse sensor locations are shown in Figs. 5(c) and 6(c), respectively. In order to make the data match low-cost IR sensory data, we model the sensor response from the pixel intensity values to reflect the limitations of their sensing capabilities. To do this, let $v(m, n, t) \in [0, 1]$ for $t \in \{1, \dots, T\}$ be a pixel intensity value at location m, n at time step t . We use the following relation to model a low-cost sensor response:

$$x_g(m, n) = \frac{1}{T} \sum_{t=1}^T I(v(m, n, t) \geq Th) \quad (14)$$

where I is the 0-1 indicator function and Th is a certain threshold. Equation (14) accounts for the low accuracy and the low temporal resolution of an IR sensor.

We then randomly generate coordinate set $\mathcal{S}$ and sample the locations $\{x_g(m, n) \mid (m, n) \in \mathcal{S}\}$ to generate our input data. It is worth mentioning that $|\mathcal{S}| \ll N \times N$, where $N \times N$ is the size of the original image x_g . We then feed these sparse images, along with constraints masks to our deep framework to try and retrieve the full image x_g . In our experiments, we set $Th = 16/255$ and $T = 16$.

IV. EXPERIMENTAL RESULTS

A. Methane Source Localization

We trained our neural network by unrolling a few steps of the forward model in Algorithm 2. We used four DCT bandwidth parameters $\mathcal{BW} = \{4, 8, 16, 32\}$. Each inference pass is four iterations ($L = 4$) with the POCS procedure applied four times ($K = 4$). Our dampening factor β is set to 0.5. We used the same parameters for training and inference. The final output $\hat{x}$ is then optimized to minimize the following loss:

$$\mathcal{L}(\hat{x}) = \|x_g - \hat{x}\|_W - \lambda \text{SSIM}(x_g, \hat{x}) \quad (15)$$

where W is a weight matrix used to give more emphasis to the area surrounding the source location and is given mathematically by

$$W(i, j) = 1 + 10 \exp \left(-\frac{|i - i^*|^2 + |j - j^*|^2}{\sigma^2} \right) \quad (16)$$

where (i^*, j^*) are the coordinates of the source. We select $\sigma = 7$. The second term SSIM in (15) refers to the structural similarity index metric [34]. The SSIM between two image patches x and y is given by

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (17)$$

where μ_x and μ_y are the means, σ_x^2 and σ_y^2 are the variances, and σ_{xy} is the cross-correlation, and c_1 and c_2 are small constants. The hyperparameter $\lambda = 10^{-3}$, and we trained the neural network over the frames of one video for ten epochs. We construct our inputs from the pixel intensity values according to (14), where our averaging window size is 16 frames, and our threshold Th is set to 10/255. This means that the values of $x_g \in \{(0/16), (1/16), \dots, (16/16)\}$.

Source Localization Results: We assess the goodness of the output of our deep-POCS network, that is, the interpolated image, by considering how close the maxima with the largest amplitude to the source. In order to do this, we first detect all local maxima of the output of the neural network. Let $\hat{x}$ be the output of the framework. We first apply a maximum filter to $\hat{x}$

$$\hat{x}^{\max}(i, j) = \max\{\hat{x}(k, l) \mid (k, l) \in \text{NBR}(i, j)\} \quad (18)$$

where the neighborhood $\text{NBR}(i, j)$ is defined as the 7×7 grid surrounding point (i, j) . We then find all the local maxima $\mathcal{MAX} := \{(i, j) \mid \hat{x}_{i,j} = \hat{x}_{i,j}^{\max}\}$. We then select the k th largest values of the set and construct the set $\mathcal{MAX}_k$. In our case, we take the top five points. Let the source be located at (p, q) . We then define our distance-to-source metric as follows:

$$\text{dist} := \min(\{d((p, q), (m, n)) \mid (m, n) \in \mathcal{MAX}_k\}) \quad (19)$$

where d is the Euclidean distance between two 2-D points. In addition to that, we perform another pass to the deep NN while removing the constraints around the maxima points in $\mathcal{MAX}_k$. Mathematically, we modify our nonzero mask $\mathbf{M}_{\text{NZ}}$ as follows:

$$[\mathbf{M}_{\text{NZ}}^{\text{new}}]_{i,j} = \begin{cases} 0, & (i, j) \in \bigcup_{(k,l) \in \mathcal{MAX}_k} \text{NBR}(k, l) \\ [\mathbf{M}_{\text{NZ}}]_{ij}, & \text{otherwise} \end{cases} \quad (20)$$

where $\text{NBR}(k, l)$ is the neighborhood of the extreme point with location (k, l) . In our case, for each of five extreme points in $\mathcal{MAX}_k$, we eliminate the spatial constraints x_s contained in their respective 7×7 grid neighborhood. Once we obtain the eroded mask $\mathbf{M}_{\text{NZ}}^{\text{new}}$, multiply it with the sparse input $x_s^{\text{new}} := x_s \circ \mathbf{M}_{\text{NZ}}^{\text{new}}$ and feed the new input to the deep model again and obtain a new interpolated output. We then detect the maxima points and compare their distances to the source location.

TABLE III
RESULTS WITH 480 SENSOR MEASUREMENTS FOR THE DEEP
REGULARIZED POCS COMPARED TO GMM FITTING
AND TRADITIONAL POCS

Vid ID	Maxima Distance to Source				
	Deep DCT-POCS one step	Deep DCT-POCS two step	GMM	POCS DCT BW=16	POCS DCT BW=32
Exp 1	22.4	15.8	19.7	27.8	29.4
Exp 2	19.1	13.3	20.1	26.6	35.1
Exp 3	11.7	10.1	11.4	17.3	10.1
Exp 4	21.3	18.9	31.9	42.1	48.4
Exp 5	15.7	9.6	12.0	18.5	17.9
Exp 6	28.1	17.9	26.5	33.3	34.7
Exp 7	12.8	8.9	12.0	17.6	11.5

TABLE IV
RESULTS WITH 100 SENSOR MEASUREMENTS FOR THE DEEP
REGULARIZED POCS COMPARED TO GMM FITTING
AND TRADITIONAL POCS

Vid ID	Maxima Distance to Source				
	Deep DCT-POCS one step	Deep DCT-POCS two step	GMM	POCS DCT BW=16	POCS DCT BW=32
Exp 1	25.1	22.5	26.2	27.7	31.4
Exp 2	19.1	14.1	22.0	27.4	29.6
Exp 3	19.9	17.1	15.3	20.4	16.6
Exp 4	27.1	24.2	36.7	43.4	44.5
Exp 5	18.1	15.9	14.3	18.7	19.7
Exp 6	28.5	20.1	25.8	31.4	35.5
Exp 7	18.7	15.2	14.1	18.4	18.6

We also compared our deep-DCT-POCS results with a Gaussian-mixture model (GMM). In this case, we start off by selecting a number of mixtures M . We then repeat the location pair (x, y) of the sparse measurements based on the intensity of measurement at that location. In our case, since the measurements correspond to the count the pixel intensity exceeds the threshold $Th = 10/255$ in a total of 16 consecutive, these counts range from 0 to 16, and so we repeat these points in our fitting datasets accordingly. For example, if the intensity at location (x_0, y_0) is 3/16 and the intensity at location (x_1, y_1) is 10/16. We repeat the location pair (x_1, y_1) 10 times, while repeating the location (x_0, y_0) three times. Notice that if the value of $x_g(m, n)$ is 0, we then do not include that point in GMM fitting process. Once we have repeated our location pairs according to their intensities, we end up with an $d \times 2$ dataset $\mathcal{D}$, where d is the final size after repetition and now we try to find the 2-D mean vectors μ_i , the 2×2 co-variance matrices C_i and the weights of each Gaussian in the mixture model π by finding maximum likelihood location

$$\{\pi_i^*, \mu_i^*, C_i^*\}_{i=0}^{M-1} = \operatorname{argmax} \sum_{i=0}^{M-1} \pi_i \mathcal{N}(\mu_i^i, C_i | \mathcal{D}) \quad (21)$$

where $\mathcal{N}$ is a 2-D Gaussian distribution. We solve (21) using the expectation-maximization algorithm. Once the algorithm converges, we then take the locations of the means μ_i as our candidate maxima locations and construct our set $\mathcal{MAX}_k$ and find the score dist defined in (19). Finally, we compare the results with the traditional POCS algorithm as explained in Section II-A. Our results are summarized in Table III for

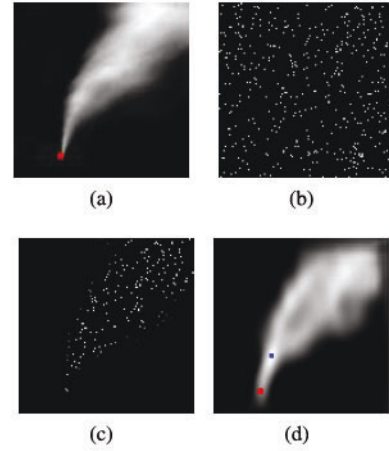


Fig. 5. Example of an input signal and the DNN-DCT-POCS with 480 input measurements. (a) Ground-truth gas leak image with the source marked in red. (b) Location of all the sensors. (c) Sparse input signal to the DNN. (d) Output of the DNN. The global maximum point is marked in blue.

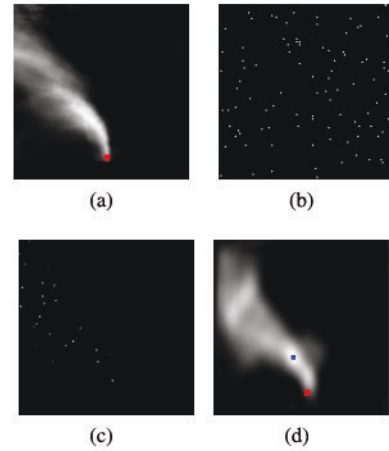


Fig. 6. Another example of an input signal and the DNN-DCT-POCS with only 100 input measurements. (a) Ground-truth gas leak image with the source marked in red. (b) Location of all the sensors. (c) Sparse input signal to the DNN. (d) Output of the DNN. The global maximum point is marked in blue.

inputs with 480 input measurements and in Table IV for inputs with 100 sensor measurements.

As one can see from Tables III and IV, the Deep DCT-POCS approach provides a better score in most cases than in both the GMM fitting and traditional POCS approaches. Furthermore, while the GMM fitting gives distances better than the Deep DCT-POCS in some examples, it is significantly worse than our method in some videos. In addition to that, the GMM will always give a higher mixture weight π_i for the distribution $\mathcal{N}(\mu_i, C_i)$ with μ_i closest to the center of mass, which is far from the peak (source) location. Visual examples are provided in Fig. 5.

B. Results: Isopropyl Alcohol Source Localization

After calibrating the chemical sensor data as mentioned in Section III-A, we downsampled the input data by a factor of 2. Therefore, our sampling rate is 1 sample/s. The original images have a spatial dimension of 18×18 . Because we use

TABLE V

AVERAGE DISTANCE BETWEEN THE LOCATION OF THE PEAK OF THE RECONSTRUCTED SIGNAL AND THE TRUE SOURCE LOCATION VERSUS THE DISTANCE BETWEEN THE TRUE SOURCE AND THE CENTROID OF THE THREE SENSOR MEASUREMENTS

Exp	avg dist	
	DNN	Centroid
Exp 1	3.14	4.49
Exp 2	2.77	2.80
Exp 3	3.89	4.82

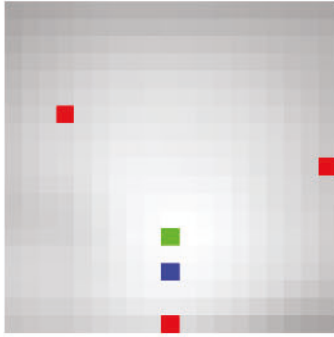


Fig. 7. Example of output image given the isopropyl alcohol sensor measurements over three locations. The sensor locations are colored in red. The source is located at the blue point, while the predicted source location (the argmax of the output image) is located at the green point.

32-point DCT transform, our input images and masks must be 32×32 pixels. We pad zeros to the original 18×18 images, and we set our zero mask $\mathbf{M}_z$ as follows:

$$[\mathbf{M}_z]_{ij} = \begin{cases} 1, & 9 \leq i \leq 24 \text{ and } 9 \leq j \leq 24 \\ 0, & \text{otherwise.} \end{cases} \quad (22)$$

This means that we limit the support to be equal to the original input image size while using a higher-resolution DCT in our POCS layers. Given the limited number of data points at each time step, we feed four consecutive time frames to our neural network in order to capture some information from the temporal behavior of the measurements at each location, that is, our input size is $32 \times 32 \times 4$, with a total number of 12 nonzero constraints.

Training the Model With Partially Known Ground Truth: Given the limited amount of measurement we have at each time step, and the fact that we do not have the full gas signal across the entire grid, this makes the problem of estimating the isopropyl alcohol source location very challenging.

In order to supervise the neural network, we create a dense “gas field” signal as follows: given the 12 input constraints, we add four additional fake measurements at the source location for each temporal map. The intensity is selected large enough to be larger than any of the measurements. We then use the POCS procedure to create an artificial dense ground-truth signal from the sparse constraints such that its peaks at each time step coincide with the true source location. We then stored these dense fields and supervised the neural network to minimize the following loss:

$$\mathcal{L}(\hat{x}) = \|\hat{x} - x_g\|_2^2 \quad (23)$$

where $\hat{x}$ is the output of the neural network, and x_g is the synthetic signal with $\text{argmax}(x_g) = (x_{\text{src}}, y_{\text{src}})$. We used two bandwidth parameters $\text{BW}_1 = 4$, and $\text{BW}_2 = 8$. The POCS procedure is run for eight iterations, while the overall procedure is run only for one iteration.

We used five experiments for training, three for validation, and three more experiments for testing our model. Our results over the test dataset are given in Table V. An example of an output image is shown in Fig. 7. As one can see, our model achieves a better distance score than predicting the source location by simply considering the center of mass. This suggests that the time-series measurements are very noisy, and simple averaging is very sensitive to these fluctuations in time series as can be seen in Fig. 3. On the other hand, the model learns robust features from the time series that are less sensitive to these fluctuations.

V. CONCLUSION

In this article, we addressed the issue of locating a leaking gas source given sparse and noisy sensor measurements. For this purpose, we proposed a data interpolation framework that combines deep neural networks with a regular POCS-based 1-D signal and two- or higher-dimensional interpolation algorithms. In particular, we utilized the iterative bandlimited interpolation algorithm, also known as the PG algorithm. Instead of using the Fourier transform we used the DCT to avoid complex numbers. The interpolation problem is ill-posed. We combined the iterative structure with a convolutional neural network to regularize the iterations to achieve reliable solutions. The overall algorithm is trained with past data.

We experimented with two different datasets at two different scales. The first example is indoor isopropyl alcohol gas leak data, which we collected using three commercially available chemical sensors. The second data corresponds to methane leaks in industrial plants that were extracted from IR videos. We tested our approach on the two datasets and we were able to interpolate the gas field spatial signal with high accuracy. We considered the local maxima of the reconstructed 2-D data as candidates for the source locations. Our approach achieved better results than GMM-based interpolation in the case of the methane data, and the center of mass-based location estimation in the case of isopropyl alcohol data.

REFERENCES

- [1] A. J. Turner et al., “Estimating global and north American methane emissions with high spatial resolution using GOSAT satellite data,” *Atmos. Chem. Phys.*, vol. 15, no. 12, pp. 7049–7069, Jun. 2015.
- [2] R. Parker et al., “Methane observations from the greenhouse gases observing SATellite: Comparison to ground-based TCCON data and model calculations,” *Geophys. Res. Lett.*, vol. 38, no. 15, pp. 1–6, Aug. 2011.
- [3] S. Jongaramrungruang, A. K. Thorpe, G. Matheou, and C. Frankenberg, “MethaNet—An AI-driven approach to quantifying methane point-source emission from high-resolution 2-D plume imagery,” *Remote Sens. Environ.*, vol. 269, Feb. 2022, Art. no. 112809.
- [4] K. C. Liddiard, “Thin-film resistance bolometer IR detectors,” *Infr. Phys.*, vol. 24, no. 1, pp. 57–64, Jan. 1984.
- [5] H. H. Hill, W. F. Siems, and R. H. St Louis, “Ion mobility spectrometry,” *Anal. Chem.*, vol. 62, no. 23, pp. 1201A–1209A, 1990.

- [6] M. A. Ryan *et al.*, "Monitoring space shuttle air quality using the jet propulsion laboratory electronic nose," *IEEE Sensors J.*, vol. 4, no. 3, pp. 337–347, Jun. 2004.
- [7] C. Yang, P. Cronin, A. Agambayev, S. Ozev, A. E. Cetin, and A. Orailoglu, "A crowd-based explosive detection system with two-level feedback sensor calibration," in *Proc. 39th Int. Conf. Comput.-Aided Des.*, Nov. 2020, pp. 1–9.
- [8] A. Vergara, S. Vembu, T. Ayhan, M. A. Ryan, M. L. Homer, and R. Huerta, "Chemical gas sensor drift compensation using classifier ensembles," *Sens. Actuators B, Chem.*, vol. 166, pp. 320–329, May 2012.
- [9] K.-M. Chang, C.-T. Chang, K.-Y. Chao, and C.-H. Lin, "A novel pH-dependent drift improvement method for zirconium dioxide gated pH-ion sensitive field effect transistors," *Sensors*, vol. 10, no. 5, pp. 4643–4654, May 2010.
- [10] D. Ma, J. Deng, and Z. Zhang, "Comparison and improvements of optimization methods for gas emission source identification," *Atmos. Environ.*, vol. 81, pp. 188–198, Dec. 2013.
- [11] A. P. Ravikumar, J. Wang, and A. R. Brandt, "Are optical gas imaging technologies effective for methane leak detection?" *Environ. Sci. Technol.*, vol. 51, no. 1, pp. 718–724, Jan. 2017.
- [12] Z. Yong, Z. Liyi, H. Jianfeng, B. Zhe, and Y. Yi, "An indoor gas leakage source localization algorithm using distributed maximum likelihood estimation in sensor networks," *J. Ambient Intell. Hum. Comput.*, vol. 10, no. 5, pp. 1703–1712, May 2019.
- [13] D. Ma, J. Gao, Z. Zhang, H. Zhao, and Q. Wang, "Locating the gas leakage source in the atmosphere using the dispersion wave method," *J. Loss Prevention Process Industries*, vol. 63, Jan. 2020, Art. no. 104031.
- [14] P. B. Jónsson, J. Wang, and J. Kim, "Scalar field reconstruction based on the Gaussian process and adaptive sampling," in *Proc. 14th Int. Conf. Ubiquitous Robots Ambient Intell. (URAI)*, Jun. 2017, pp. 442–445.
- [15] M. Asenov, M. Rutkauskas, D. Reid, K. Subr, and S. Ramamoorthy, "Active localization of gas leaks using fluid simulation," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1776–1783, Apr. 2019.
- [16] M. Hutchinson, H. Oh, and W.-H. Chen, "A review of source term estimation methods for atmospheric dispersion events using static or mobile sensors," *Inf. Fusion*, vol. 36, pp. 130–148, Jul. 2017.
- [17] A. Papoulis, "A new algorithm in spectral analysis and band-limited extrapolation," *IEEE Trans. Circuits Syst.*, vol. CS-22, no. 9, pp. 735–742, Sep. 1975.
- [18] D. C. Youla and H. Webb, "Image restoration by the method of convex projections: Part 1 theory," *IEEE Trans. Med. Imag.*, vol. MI-1, no. 2, pp. 81–94, Oct. 1982.
- [19] D. Huang and H. Leung, "Reconstruction of drifting sensor responses based on Papoulis–Gerchberg method," *IEEE Sensors J.*, vol. 9, no. 5, pp. 595–604, May 2009.
- [20] P. L. Combettes, "The foundations of set theoretic estimation," *Proc. IEEE*, vol. 81, no. 2, pp. 182–208, Feb. 1993.
- [21] P. Combettes, "The convex feasibility problem in image recovery," in *Advances in Imaging and Electron Physics*, vol. 95. Amsterdam, The Netherlands: Elsevier, 1996, pp. 155–270.
- [22] R. W. Gerchberg, "Super-resolution through error energy reduction," *Opt. Acta, Int. J. Opt.*, vol. 21, no. 9, pp. 709–720, 1974.
- [23] I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm," *IEEE Trans. Signal Process.*, vol. 45, no. 3, pp. 600–616, Mar. 1997.
- [24] A. Lent and H. Tuy, "An iterative method for the extrapolation of band-limited functions," *J. Math. Anal. Appl.*, vol. 83, no. 2, pp. 554–565, Oct. 1981.
- [25] B. G. Salomon and H. Ur, "Accelerated iterative band-limited extrapolation algorithms," *IEEE Signal Process. Lett.*, vol. 11, no. 11, pp. 871–874, Nov. 2004.
- [26] C.-Y. Hsu and T.-M. Lo, "Improved Papoulis–Gerchberg algorithm for restoring lost samples," in *Proc. 5th IEEE Int. Symp. Signal Process. Inf. Technol.*, Dec. 2005, pp. 717–721.
- [27] C. Chamzas and W. Xu, "An improved version of Papoulis–Gerchberg algorithm on band-limited extrapolation," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-32, no. 2, pp. 437–440, Apr. 1984.
- [28] S. Diamond, V. Sitzmann, F. Heide, and G. Wetzstein, "Unrolled optimization with deep priors," 2017, *arXiv:1705.08041*.
- [29] G. Ongie, A. Jalal, C. A. Metzler, R. G. Baraniuk, A. G. Dimakis, and R. Willett, "Deep learning techniques for inverse problems in imaging," *IEEE J. Sel. Areas Inf. Theory*, vol. 1, no. 1, pp. 39–56, May 2020.
- [30] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. 27th Int. Conf. Int. Conf. Mach. Learn.*, 2010, pp. 399–406.
- [31] J. Watson, K. Ihokura, and G. S. Coles, "The tin dioxide gas sensor," *Meas. Sci. Technol.*, vol. 4, no. 7, p. 711, 1993.
- [32] F. Erden, E. B. Soyer, B. U. Toreyin, and A. E. Cetin, "VOC gas leak detection using pyro-electric infrared sensors," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2010, pp. 1682–1685.
- [33] B. D. Lucas, "An iterative image registration technique with an application to stereo vision (darpa)," in *Proc. DARPA Image Understand. Workshop*, Apr. 1981.
- [34] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Proc. 37th Asilomar Conf. Signals, Syst. Comput.*, vol. 2, Jul. 2003, pp. 1398–1402.

Diaa Badawi received the M.Sc. degree in electrical engineering from Bilkent University, Ankara, Turkey, in 2018, and the Ph.D. degree in electrical engineering from the University of Illinois at Chicago, Chicago, IL, USA, in 2022.

His research interests include intelligent monitoring systems, energy-efficient implementations of neural networks, and machine learning algorithms.

Ishaan Bassi received the M.Sc. degree in electrical and electronics engineering from Arizona State University (ASU), Tempe, AZ, USA, in 2020, where he is currently pursuing the Ph.D. degree.

His research interests include solid-state physics and photonics.

Sule Ozev (Member, IEEE) received the Ph.D. degree in computer science and engineering from the University of California at San Diego, San Diego, CA, USA, in 2002.

She is currently an Associate Professor with the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ, USA. Her research interests include self-test and self-calibration for wireless transceivers, analysis, and mitigation of process variations for mixed-signal and digital circuits, fault-tolerant and reconfigurable heterogeneous systems, and mixed-signal circuit testing.

A. Enis Cetin (Fellow, IEEE) received the B.Sc., M.S.E., and Ph.D. degrees in systems engineering from the Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia, PA, USA, in 1984, 1986, and 1987, respectively. He studied electrical engineering with Orta Dogu Teknik Universitesi (METU), Ankara, Turkey.

From 1987 to 1989, he was an Assistant Professor in Electrical Engineering with the University of Toronto, Toronto, ON, Canada. He is on leave from Bilkent University, Ankara. From 2016 to 2017, he was a Visiting Professor with the University of California at San Diego, San Diego, CA, USA. He is currently a Research Professor with the University of Illinois at Chicago, Chicago, IL, USA. He is one of the founders of GrandEye, London, U.K., that produces omnidirectional cameras.