Multimodal Unrolled Robust PCA for Background Foreground Separation

Spencer Markowitz, Corey Snyder, Yonina C. Eldar, *Fellow, IEEE*, and Minh N. Do, *Fellow, IEEE*

***Abstract*—Background foreground separation (BFS) is a popular computer vision problem where dynamic foreground objects are separated from the static background of a scene. Typically, this is performed using consumer cameras because of their low cost, human interpretability, and high resolution. Yet, cameras and the BFS algorithms that process their data have common failure modes due to lighting changes, highly reflective surfaces, and occlusion. One solution is to incorporate an additional sensor modality that provides robustness to such failure modes. In this paper, we explore the ability of a cost-effective radar system to augment the popular Robust PCA technique for BFS. We apply the emerging technique of algorithm unrolling to yield real-time computation, feedforward inference, and strong generalization in comparison with traditional deep learning methods. We benchmark on the RaDICaL dataset to demonstrate both quantitative improvements of incorporating radar data and qualitative improvements that confirm robustness to common failure modes of image-based methods.**

***Index Terms*—Radar, background foreground separation, algorithm unrolling, ISTA.**

I. Introduction

Background foreground separation (BFS) is a fundamental task for many computer vision algorithms where dynamic foreground components are separated from the static background of a given scene. Successful BFS enables applications in intelligent surveillance such as vehicular traffic monitoring, industrial manufacturing, and human activity recognition [1]. A wide variety of approaches to BFS exist in the literature. Subspace methods and deep learning have emerged as the dominant techniques in the past decade due to their superior performance on popular benchmark datasets [2], [3]. Other techniques include statistical methods, fuzzy models, and cluster models. For a recent review, see [4].

In this work, we operate in the unsupervised BFS problem setting where no hand-labeled data is available during training and side-information from an additional sensing modality is used alongside camera data. To the best of our knowledge, this is the first such work in BFS that incorporates radar sensing into a BFS algorithm. Automotive/consumer radar sensing has recently seen a great deal of interest due to its affordability, compact size, and ability to sense in conditions where cameras perform poorly. In the context of BFS, radar does not detect some undesirable foreground components that cameras capture. Examples include shadows, changes in lighting, reflections and digital screens as depicted in the Venn diagram in Fig. 1. Moreover, compared to its camera counterpart, detecting moving targets in radar data is much simpler. This is achieved by measuring subtle changes in the phase of the received signal. This allows radar to detect salient motion in as little as one frame depending on the type of radar. A comparison between camera and radar data of the same scene is shown in Fig. 2. Unlike cameras, radar sensing's shortcomings include low angular resolution, specularity, and multipath, the first two of which can be seen in Fig. 2b.

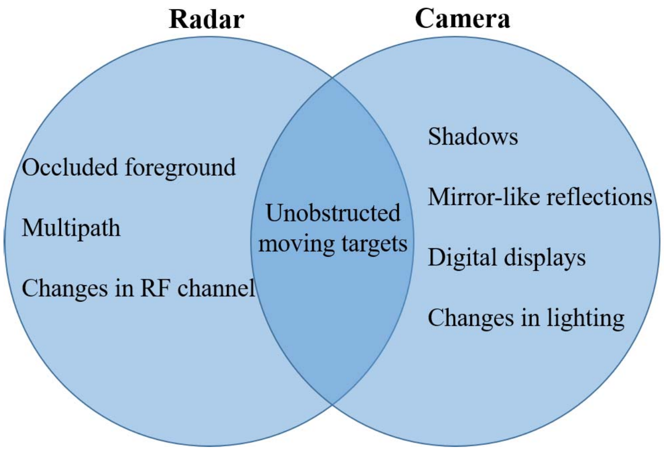


Fig. 1. Examples of targets or phenomena that may potentially be detected as foreground for radar and camera sensors.

In designing our algorithm, we make the practical considerations for (1) real-time computation, (2) robustness to unseen data, and (3) cost-effectiveness. Towards the first two points, we leverage the advantages of both subspace and deep learning models via the emerging technique of algorithm unrolling. First proposed in [5] for sparse coding, an iterative algorithm is *unrolled* or *unfolded* by representing the k'th iteration as the k'th layer in a feedforward network. The result of the k'th layer is fed as the input to layer $k+1$ where common operations

Manuscript received August 13, 2021; revised February 20, 2022 and April 12, 2022; accepted April 16, 2022. Date of publication May 11, 2022; date of current version May 18, 2022. This work was supported in part by the Sandia National Laboratory under LDRD Project and in part by Texas Instruments. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Raja Giryes. *(Spencer Markowitz and Corey Snyder contributed equally to this work.) (Corresponding author: Spencer Markowitz.)*

Spencer Markowitz and Corey Snyder are with the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Champaign, IL 61820 USA (e-mail: samarko2@illinois.edu).

Yonina C. Eldar is with the Department of Mathematics and Computer Science, Weizmann Institute of Science, Rehovot 7610001, Israel.

Minh N. Do is with the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Champaign, IL 61820 USA, and also with the VinUni-Illinois Smart Health Center, VinUniversity, Hanoi 907012, Vietnam.

Digital Object Identifier 10.1109/TIP.2022.3172851

of iterative algorithms such as shrinkage operators function as the non-linearities in traditional deep nets, e.g. ReLU. Unrolled networks have been shown to achieve the same performance as their iterative counterparts using dramatically fewer layers. This means real-time computation is possible on both seen and unseen data without sacrificing performance. Furthermore, compared to state of the art deep nets, unrolled neural networks often use far fewer parameters, require less training data, and maintain a high level of interpretability due to the structure imposed by its accompanying "white box" iterative algorithm [6].

In addition to sparse coding, algorithm unrolling has also been used to tackle a wide variety of problems employing well studied algorithms and supplementing key assumptions with data-driven techniques. Examples of unrolling algorithms into feedforward networks include image deblurring [7], phase retrieval [8], channel estimation [9], and clutter suppression in ultrasound [10].

In this paper, we extend the work of CORONA [10], which is an unrolled Robust PCA technique. We combine radar side-information with camera data into the Robust PCA objective to re-weight the penalty of the sparse foreground. We present the ISTA algorithm for this radar-modified objective and refer to this procedure as RISTA. We then unroll RISTA into a feedforward convolutional neural network we call Radar Unrolled Shrinking and Thresholding Incorporating Convolutions, or RUSTIC. To ensure our method is cost-effective and practical, we use frequency-modulated continuous wave (FMCW) radar for our experimentation. FMCW radar is low cost (the Texas Instruments IWR1443 we use costs $12 USD) and small in size (can be put on a single PCB). FMCW radar can transmit *and* receive simultaneously thus allowing detection of targets at very close range. We perform quantitative evaluation on the RaDICaL dataset[1] [11] and demonstrate that RUSTIC delivers competitive and sometimes superior performance to its iterative counterpart on both seen and unseen data while enabling real-time computation. We also compare RUSTIC to the CORONA model that only utilizes camera data and a conventional deep learning segmentation model in the U-Net [12]. We show a clear improvement in quantitative performance by incorporating radar side-information for shallower unrolled models and demonstrate these unrolled models generalize far better to unseen scenes than the U-Net while using orders of magnitude fewer parameters. Furthermore, we provide qualitative examples illustrating the effectiveness of both the camera and radar modalities to correct errors from one another through the RUSTIC framework.

The rest of the paper is organized as follows. In Section II, we discuss prior work in BFS involving subspace and deep learning approaches. Section III defines our problem setting and motivates the incorporation of radar data into the RPCA objective to form our iterative RISTA algorithm. In Section IV, we explain how RISTA is unrolled into our RUSTIC model. Section V details our quantitative and qualitative experimental results for RUSTIC and related works on the RaDICaL dataset. Finally, we conclude in Section VI and provide suggestions for future work using RUSTIC and sensor fusion in BFS.

II. Related Work

A. Unsupervised Subspace Methods

Unsupervised subspace methods seek to decompose an image sequence into the sum of a low-rank background and sparse foreground. Robust PCA (RPCA) [13] is one highly influential method which solves the convex Principle Component Pursuit (PCP) program to perform this separation. While effective in many settings, PCP can take hundreds of iterations to converge using popular solvers such as ISTA or ADMM, and subspaces must be recomputed when new data is made available. These shortcomings have been explored in papers proposing faster optimization algorithms [14] as well as with more significant changes to the algorithm to enable real-time RPCA [15]–[19].

An additional shortcoming of pure subspace methods is that they do not take into account the spatial-temporal constraints of real-world moving objects. For example, with a sufficiently high video frame rate, one can assume foreground objects will not move drastically between consecutive frames. In [20]–[22], this constraint is addressed in the objective function in order to estimate a foreground more robust to noise and identify dynamic backgrounds like moving water.

Another method for improving the performance of subspace methods, as done in this paper, is to include an additional sensor. Much of the published research in this area uses depth information in the form of RGB-D data because of ease of use with RGB data, especially when the data comes from a single device. While [23], [24] do demonstrate improvements in performance with RGB-D data compared to RGB data alone, depth data is rather limited in its application because of its restrictive maximum sensing depth and its ability to accurately estimate depth in nonideal conditions such as low lighting. Unlike depth sensors, such problems do not curb the performance of radar.

In addition to these drawbacks, subspace methods are also sensitive to changes in lighting and camera alignment. For example, a small translation of the camera defines a completely new low-rank subspace for the background. Because there is no feature learning in subspace methods, such subtle changes cannot be recognized or properly discarded without explicitly imposing greater structure on the subspace model.

B. Supervised/Unsupervised Deep Learning Methods

Supervised deep learning techniques [3], [25] have been shown to address the aforementioned limitations and even provide human-level performance on supervised learning benchmarks like CDnet14 [26] and Scene Background Initialization 2015 [27]. With hand-labeled ground-truth examples, Convolutional Neural Networks (CNNs) learn rich features that focus on salient changes in a scene and are robust to changes like the

[1] The dataset can be accessed at https://databank.illinois.edu/datasets/IDB-3289560

aforementioned camera shift due to the translation-invariance of the convolution operator. Deep learning algorithms often require an expensive fitting or training process like subspace methods; however, they are able to be deployed immediately on unseen data without the re-fitting that subspace methods require. Example models include FgSegNet [28], CascadeCNN [29], the Deep Difference Network [30], BSUV-Net [31], and [32] which leverages Graph CNNs. Furthermore, [33] and [34] focus on efficient implementations of using deep learning for such BFS tasks. Additional research in deep learning has also addressed the task of intelligently fusing early stage and minimally processed radar data with RGB data [11], [35].

The key shortcoming of deep learning methods is that they require expensive pixel-level ground truths for hundreds of images. These deep CNNs typically have on the order of millions of learnable parameters. Thus, when limited ground truths are available, they are prone to overfitting and poor generalization to unseen data. Unsupervised deep learning approaches have been suggested to alleviate the labeling burden; however, they lag considerably behind their supervised counterparts [36]. Recent works including BSUV-Net [31] and BSUV-Net 2.0 [37] have improved the performance of supervised models on unseen scenes by designing augmentation policies, but still require detailed annotations on many training scenes.

Accordingly, in this paper we seek to leverage the strong capabilities of modern deep learning methods with the structure of iterative subspace methods in order to achieve an unsupervised solution to BFS.

III. Sensor Fusion for BFS

A. Problem Setup

We consider the scenario where M radar frames and M camera frames, each separated by Δt, observe the same scene and are synchronized in time. Using both the camera and radar data, we aim to separate the camera data into its background and foreground components. Let $\mathbf{D}_m \in \mathbb{R}^{H\times W}$ be a single frame in our video sequence for a given scene. For the radar data, we assume the radar transmits a constant amplitude sawtooth waveform such that for a given chirp interval, $0 < t < T$, the frequency can be expressed as $f_c + Bt$ where f_c is the starting frequency and B is the chirp slope. This yields the following transmitted waveform

$$S_{tx} = A_{tx}\cos\left(2\pi\left(f_c t + \frac{B}{2}t^2\right)\right) \tag{1}$$

where A_{tx} is the signal amplitude. After the transmitted signal is reflected from a target back to the radar's receivers, the signal is subsequently mixed with the transmitted signal and low-pass filtered to obtain the intermediate frequency (IF) signal. The IF signal is then sampled N_s times during the chirp's interval. We also assume that $T \ll \Delta t$ allowing us to include multiple radar chirps in each radar frame. Accordingly, we consider one complete radar frame to contain N_c sequential chirps each received and sampled at N_a receivers. This results in a single frame of radar data taking the form $\mathbf{R}_m \in \mathbb{C}^{N_s\times N_a\times N_c}$.

Next, we follow the RPCA [13] subspace method for BFS closely and seek to separate our camera data $\mathbf{D}$ into its low-rank and sparse components, $\mathbf{L}$ and $\mathbf{S}$, respectively. We accomplish this by first vectorizing each frame of $\mathbf{D}$ such that $\mathbf{D}$, $\mathbf{L}$ and $\mathbf{S}$ all belong to $\mathbb{R}^{HW\times M}$. The low-rank+sparse decomposition objective is commonly stated as follows:

$$\min_{\mathbf{L},\mathbf{S}} \ \mathrm{rank}(\mathbf{L}) + ||\mathbf{S}||_0, \quad \text{s.t. } \mathbf{D} = \mathbf{L} + \mathbf{S}. \tag{2}$$

Since this program is non-convex, we use the popular convex relaxation

$$\min_{\mathbf{L},\mathbf{S}} \ ||\mathbf{L}||_* + \lambda||\mathbf{S}||_1, \quad \text{s.t. } \mathbf{D} = \mathbf{L} + \mathbf{S} \tag{3}$$

where $||\cdot||_*$ and $||\cdot||_1$ are the nuclear and l_1 norms, respectively. In the following section, we describe how radar side-information can be incorporated into (3) and then present the resulting iterative solver that will become the foundation for our unrolled feedforward network.

B. Incorporating Radar

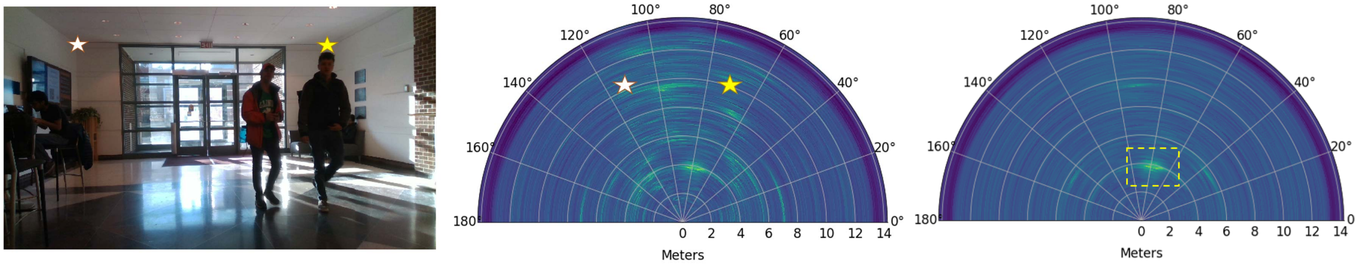
One of the many advantages that radar systems have over cameras is their ability to easily localize motion within a frame and remove any static clutter. Here, we perform this relatively simple operation first and then incorporate the clutter-free radar return in the BFS of the camera data. We assume that the clutter-free radar returns can provide useful information on where foreground is likely to exist in the camera data. Although the camera's low rank component is a function of its own sparse component, we do not use the radar data directly with the low rank prediction. We make this choice because the radar data does not provide informative cues like it can for the foreground component. For example, specularity and the physical properties of common construction materials can prevent portions of walls from being detected. These impacts are seen in Fig. 2b where the walls have many undetected patches. Conversely, moving targets are consistently detected as depicted in Fig. 2b and 2c after clutter suppression. Thus, we do not use the extracted static clutter and choose only to use the moving targets' radar reflections to convey the locations and associated likelihoods of foreground in the camera images.

Intuitively, we seek to modify the RPCA objective in (3) to make sparse foreground contributions less costly in regions where the clutter-free radar return is high and make contributions in regions where it is low more difficult to admit. We therefore modify (3) and suggest solving

$$\min_{\mathbf{L},\mathbf{S}} ||\mathbf{L}||_* + \lambda||\mathbf{S}\circ\mathcal{F}(\mathbf{R})||_1, \quad \text{s.t. } \mathbf{D} = \mathbf{L} + \mathbf{S} \tag{4}$$

where $\mathcal{F}(\cdot) : \mathbb{C}^{M\times N_s\times N_a\times N_c} \mapsto \mathbb{R}^{HW\times M}$ maps the radar data to a clutter-free, real-valued weight matrix and $\circ$ is the element-wise Hadamard product. We will make the radar processing pipeline that forms $\mathcal{F}(\cdot)$ concrete in the following subsection.

The program in (4) can be solved efficiently using a number of solvers such as ADMM or ISTA. We choose to closely follow the derivation presented in [10] and select ISTA with the addition of the radar side-information. We introduce the equality constraint in (3) and (4) from the objective function as



(a) RGB Image (b) Range-azimuth heatmap with clutter (c) Range-azimuth heatmap without clutter

Fig. 2. Example range-azimuth heatmaps for a particular scene (scene B in the appendix). As a reference, there are two stars at the back corners of the lobby in both (a) and (b). Notice in (b) the wall is not represented by a continuous line of reflections. Figure (c) shows the resulting clutter-free heatmap using the methods described in Sec. III-C. With the clutter suppression, the humans (highlighted by the bounding box) become much more visible within the heatmap.

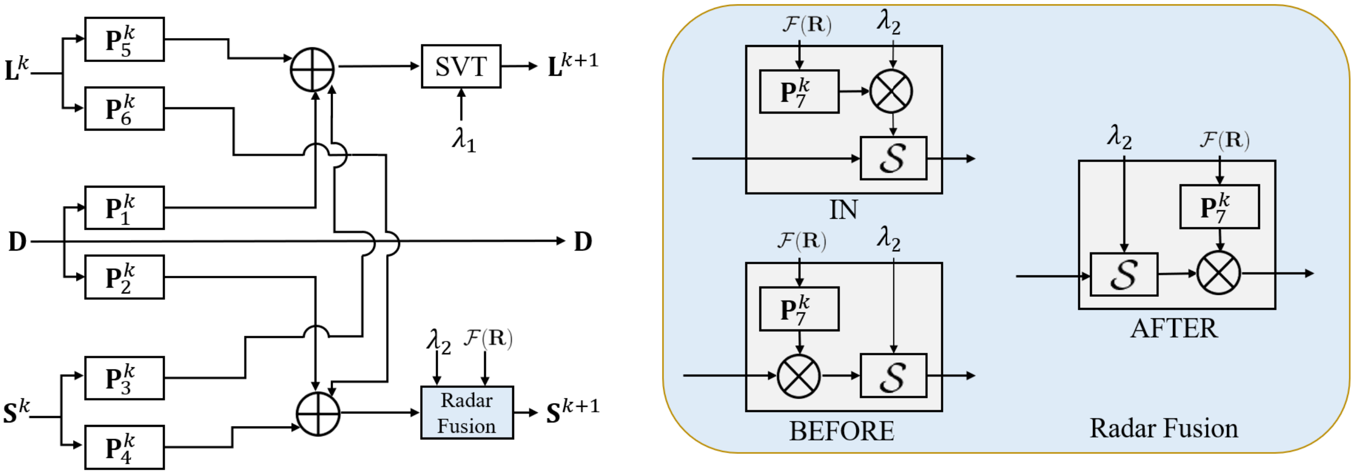


Fig. 3. On the left is a depiction of a single layer of our RUSTIC architecture. On the right, we show the three options we consider for how to incorporate the radar into the unrolled network based on Alg. 1. While incorporating the radar *in* the shrinkage operator is the most accurate interpretation of Alg. 1, we also experiment with two looser interpretations, namely using the radar *before* and *after* the shrinkage operator.

a quadratic penalty and add measurement matrices $\{\mathbf{H}_i\}_{i=1}^2$ for the low-rank and sparse components. We also multiply the radar input with its own measurement matrix, $\mathbf{H}_3$ to account for proper scaling and filtering as it corresponds with the camera data. This results in the problem

$$\min_{\mathbf{L},\mathbf{S}} ||\mathbf{D}-\mathbf{H}_1\mathbf{L}-\mathbf{H}_2\mathbf{S}||_F^2+\lambda_1||\mathbf{L}||_*+\lambda_2||\mathbf{S}\circ\mathbf{H}_3\mathcal{F}(\mathbf{R})||_1 \tag{5}$$

where $\lambda_1, \lambda_2 > 0$. We set λ_1 and λ_2 according to the conditions used in the original RPCA paper [13]. The choice of measurement matrices $\{\mathbf{H}_i\}_{i=1}^3$ is application-dependent. We make the simplifying assumption that each measurement matrix is identity since we have no prior knowledge of a more informed choice. We still leave the operators in place since they will be further abstracted to enrich our unrolled model described in Section IV.

Following [10], the modified radar-ISTA, *RISTA*, is shown in Alg. 1 where $\mathbf{X}^H$ is the Hermitian transpose, $\mathbf{I}$ is an appropriately sized identity matrix, $\mathcal{S}_\tau(x) := \text{sgn}(x)\max(|x|-\tau, 0)$, and $\text{SVT}_\tau(\mathbf{X}) := \mathbf{U}\mathcal{S}_\tau(\Sigma)\mathbf{V}^T$ where $\mathbf{X}=\mathbf{U}\Sigma\mathbf{V}^T$ is the singular value decomposition of $\mathbf{X}$. The constant μ represents the step size for the proximal gradient operator and is given by the spectral norm of $\mathbf{H}^H\mathbf{H}$ where

Algorithm 1 RISTA for Minimizing (5)

```
Input: $\mathbf{D}, \mathcal{F}(\mathbf{R}), \lambda_1, \lambda_2 > 0$
Output: $\mathbf{L}^{K_{\max}}, \mathbf{S}^{K_{\max}}$
Initialize: $\mathbf{S}^0 = \mathbf{L}^0 = \mathbf{0}, k = 0$
while not converged or $k < K_{max}$ do
    $\mathbf{G}_1^k = \left(\mathbf{I}-\mu\mathbf{H}_1^H\mathbf{H}_1\right)\mathbf{L}^k - \mathbf{H}_1^H\mathbf{H}_2\mathbf{S}^k + \mathbf{H}_1^H\mathbf{D}$
    $\mathbf{G}_2^k = \left(\mathbf{I}-\mu\mathbf{H}_2^H\mathbf{H}_2\right)\mathbf{S}^k - \mathbf{H}_2^H\mathbf{H}_1\mathbf{L}^k + \mathbf{H}_2^H\mathbf{D}$
    $\mathbf{L}^{k+1} = \text{SVT}_{\mu\lambda_1}(\mathbf{G}_1^k)$
    $\mathbf{S}^{k+1} = \mathcal{S}_{\mu\lambda_2\mathbf{H}_3\mathcal{F}(\mathbf{R})}(\mathbf{G}_2^k)$
    $k \leftarrow k+1$
end
```

$$\mathbf{H} = \begin{bmatrix}\mathbf{H}_1\\ \mathbf{H}_2\end{bmatrix}. \tag{6}$$

When computing $\mathbf{S}^{k+1}$ in Alg. 1, we use a different threshold in the shrinkage operator for each column based on the processed radar data $\mathcal{F}(\mathbf{R})$ along with the terms μ, λ_2, and $\mathbf{H}_3$. As such, for frame m, row h, and column w the threshold in the shrinkage operator can be taken as the $hW+w$'th entry in $\mu\lambda_2\mathbf{H}_3\mathcal{F}(\mathbf{R}_m) \in \mathbb{R}^{HW}$.

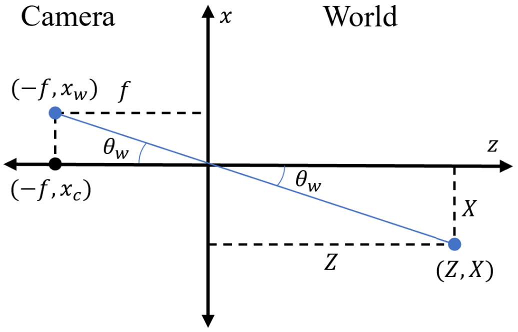


Fig. 4. Depiction of a pinhole camera model.

C. Radar Processing

As mentioned in Section III-B, radars allow for easy clutter suppression compared to cameras. Unlike cameras, clutter suppression for radar only requires data from a single radar frame and, as a result, each radar frame can be processed independently. We will drop the m subscript and let $\mathbf{R} := \mathbf{R}_m \in \mathbb{C}^{N_s \times N_a \times N_c}$ for notational brevity in this section.

The first step in processing the raw samples from the radar is to compute the range of the reflections in each chirp from each antenna [11]. This is performed by taking the Fast Fourier Transform (FFT) along each chirp's IF signal, or along the first dimension of $\mathbf{R}$ such that

$$\hat{\mathbf{R}}_{k_s n_a n_c} = \mathrm{FFT}_{n_s}\{\mathbf{R}_{n_s n_a n_c}\} \tag{7}$$

where k_s is a frequency domain index. After the range information is computed, we remove the clutter by computing the mean across the chirps within the frame and then subtract it from the range information data

$$\mu_{k_s n_a} = \frac{1}{N_c}\sum_{n_c=1}^{N_c} \hat{\mathbf{R}}_{k_s n_a n_c} \tag{8}$$

$$\tilde{\mathbf{R}}_{k_s n_a n_c} = \hat{\mathbf{R}}_{k_s n_a n_c} - \mu_{k_s n_a}. \tag{9}$$

Once the static clutter is removed, we use the multiple antennas to determine the received power at each bearing. The data used for our experiments was taken with a 1D uniform linear array with resolution along the azimuthal axis because of the low resolution in the elevation axis. We use the standard pinhole camera model with focal length f and camera center column x_c e.g. $x_c = x_{W/2}$ to map a point in space to its horizontal pixel coordinates. As such, a point with horizontal coordinate X and depth Z is mapped to

$$(X, Z) \mapsto x_w := \frac{fX}{Z} + x_c \tag{10}$$

as shown in Fig. 4. Since our data assumes the radar and camera are coplanar and stacked vertically on top of each other, we may compute the bearing θ_w corresponding to each of the image's columns x_w for $w = 1, \ldots, W$ as

$$\theta_w = \arctan\left(\frac{x_w - x_c}{f}\right). \tag{11}$$

To compute the power at each θ_w, we use Minimum Variance Distortionless Response beamforming [38]. This is accomplished by first computing the covariance matrix Σ_{k_s} for each range slice $\tilde{\mathbf{R}}_{k_s}$ such that

$$\Sigma_{k_s} = \frac{1}{N_c}\tilde{\mathbf{R}}_{k_s}\tilde{\mathbf{R}}_{k_s}^H. \tag{12}$$

The received power at each range and angle is computed as

$$\mathbf{P}(k_s, \theta_w) = \frac{1}{\mathbf{a}(\theta_w)\Sigma_{k_s}^{-1}\mathbf{a}^H(\theta_w)} \tag{13}$$

where the steering vector $\mathbf{a}(\theta_w)$ simplifies to $[1, e^{-j\pi\sin(\theta_w)}, \ldots, e^{-j(N_a-1)\pi\sin(\theta_w)}]$ because the antennas are spaced half a wavelength apart. We then take the $\log(\cdot)$ in (13) since the data often spans many orders of magnitude. Finally, we sum over the range dimension because the camera data lacks any depth information:

$$\mathbf{P}(\theta_w) = \sum_{k_s=1}^{N_s} \log[\mathbf{P}(k_s, \theta_w)]. \tag{14}$$

We use the associated bearing θ_w for each column in the image data according to (11) to compute the received power at each column and form $\mathbf{P} \in \mathbb{R}^W$. Although $\mathbf{P}$ has no elevation data and can therefore be expressed as a 1D vector, we assert its size to be the same as each camera image so it can be multiplied elementwise with the camera data in Alg. 1. Thus, we expand $\mathbf{P}$ (with a slight abuse of notation) to be of shape (H, W) by making each row identical. To summarize, $\mathbf{P}$ represents the result of $\mathcal{F}(\mathbf{R})$ that is computed independently for each radar frame $m \in [M]$.

IV. Unrolled Network With Radar

A. Model Architecture

An iterative algorithm can be modeled as an unrolled neural network where the k'th layer corresponds to the k'th iteration [6], [39]. As in [10], we replace matrix multiplication using $\mathbf{H}_{\{1,2\}}$ with 2D convolutional layers $\{\mathbf{P}_i^k\}_{i=1}^6$ as well as multiplication with $\mathbf{H}_3$ with 1D convolution layers $\mathbf{P}_7^k$, and learn λ_i^k for the shrinkage and SVT operations. The choice of 2D and 1D convolutional operators (as opposed to fully connected layers) promotes spatial coherence, reduces the number of learnable parameters, and provides the network with the desirable property of translation invariance. Altogether, Alg. 1 can be represented as a multi-layer feedforward network with each layer being described by

$$\begin{aligned} \mathbf{L}^{k+1} &= \mathrm{SVT}_{\lambda_1^k}\{\mathbf{P}_5^k * \mathbf{L}^k + \mathbf{P}_3^k * \mathbf{S}^k + \mathbf{P}_1^k * \mathbf{D}\} \\ \mathbf{S}^{k+1} &= \mathcal{S}_{\lambda_2^k \mathbf{P}_7^k * \mathcal{F}(\mathbf{R})}\{\mathbf{P}_6^k * \mathbf{L}^k + \mathbf{P}_4^k * \mathbf{S}^k + \mathbf{P}_2^k * \mathbf{D}\} \end{aligned} \tag{15}$$

with $*$ being the convolution operator and $\mathbf{S}^0 = \mathbf{L}^0 = \mathbf{0}$. The shrinkage operator here follows the same notation as in Alg. 1 where the threshold for layer k, frame m, row h, and column w is determined by the $hW + w$'th entry in $\lambda_2^k \mathbf{P}_7^k * \mathcal{F}(\mathbf{R}_m)$. The image data, $\mathbf{D}$, $\mathbf{L}$, and $\mathbf{S}$, are of shape (M, H, W). In order to perform the SVT operation, we vectorize the result of the convolution and addition operations for the updated low-rank component and stack along the second dimension to yield a shape of (HW, M). We undo this procedure after SVT is performed. It is important to note here that while the iterative algorithm uses the same thresholds and measurement matrices for all iterations, the unrolled model learns different filters and thresholds for each layer. This is a unique advantage of unrolled networks with respect to their iterative counterparts.

We also experiment with looser interpretations of Alg. 1 where instead of incorporating the radar directly *in* the sparse shrinkage operator as in (15), we incorporate it *before* or *after* as described in (16) and (17), respectively, and shown in Fig. 3:

$$\mathbf{S}^{k+1} = \mathcal{S}_{\lambda_2^k}\{(\mathbf{P}_7^k * \mathcal{F}(\mathbf{R})) \circ (\mathbf{P}_6^k * \mathbf{L}^k + \mathbf{P}_4^k * \mathbf{S}^k + \mathbf{P}_2^k * \mathbf{D})\}, \tag{16}$$

$$\mathbf{S}^{k+1} = \mathcal{S}_{\lambda_2^k}\{\mathbf{P}_6^k * \mathbf{L}^k + \mathbf{P}_4^k * \mathbf{S}^k + \mathbf{P}_2^k * \mathbf{D}\} \circ (\mathbf{P}_7^k * \mathcal{F}(\mathbf{R})). \tag{17}$$

We designate these three variations (*in* (15), *before* (16), and *after* (17)) as Radar Unrolled Shrinking and Thresholding Incorporating Convolutions (RUSTIC). In practice, we find the model that incorporates radar *after* the shrinkage operator performs the best. Section V will compare these three variations.

B. Model Training

To train the RUSTIC models, we first generate low-rank $\hat{\mathbf{L}}$ and sparse $\hat{\mathbf{S}}$ targets corresponding to the input $\mathbf{D}$. These targets are used to train the unrolled networks via backpropagation with a suitable loss function. The loss function we choose is the mean squared error (MSE) between the targets $\hat{\mathbf{L}}_i$, $\hat{\mathbf{S}}_i$ and the model predictions $\mathbf{L}_m$, $\mathbf{S}_m$

$$\mathcal{L}(\theta) = \frac{1}{2M}\sum_{m=1}^{M}\left(\left\|\mathbf{S}_m - \hat{\mathbf{S}}_m\right\|_F^2 + \left\|\mathbf{L}_m - \hat{\mathbf{L}}_m\right\|_F^2\right). \tag{18}$$

We use ISTA to form the targets and will describe the procedure for generating $\hat{\mathbf{L}}$ and $\hat{\mathbf{S}}$ in greater detail in Section V-A.

V. Experimentation

A. Setup

To evaluate our models, we compare the three variations of RUSTIC.[2] depicted in Fig. 3 along with a baseline model without radar (CORONA [10]) and a standard U-Net [12]. We use data from RaDICaL, a synchronized FMCW radar, depth, IMU and RGB dataset [11]. Each sequence contains images downsampled to 180 × 320 and transformed to grayscale. Both the camera and radar data are also downsampled in time so that the frame rates are 3 frames per second. As described in [11], the radar data was collected using the Texas Instruments IWR1443BOOST and used 4 receiving antennas and 2 transmitting antennas. By exploiting time division multiplexing this configuration yields 8 virtual receiving antennas in the horizontal axis.

We generate the sparse and low-rank components for the targets by solving the RPCA objective using ISTA without radar side-information. We assume identity measurement matrices $\mathbf{I} = \mathbf{H}_1 = \mathbf{H}_2$, thus $\mu = 1$, and run it for 400 iterations. By using ISTA instead of RISTA, we avoid having to tune additional hyperparameters ($\mathbf{H}_3$) that control how to properly scale the radar data for fusion with camera data. In order to evaluate models after training as well as the quality of generated ISTA targets, we hand-label binary images with each pixel labeled as either foreground or background. Only desirable foreground components such as moving humans and doors are labeled as foreground.

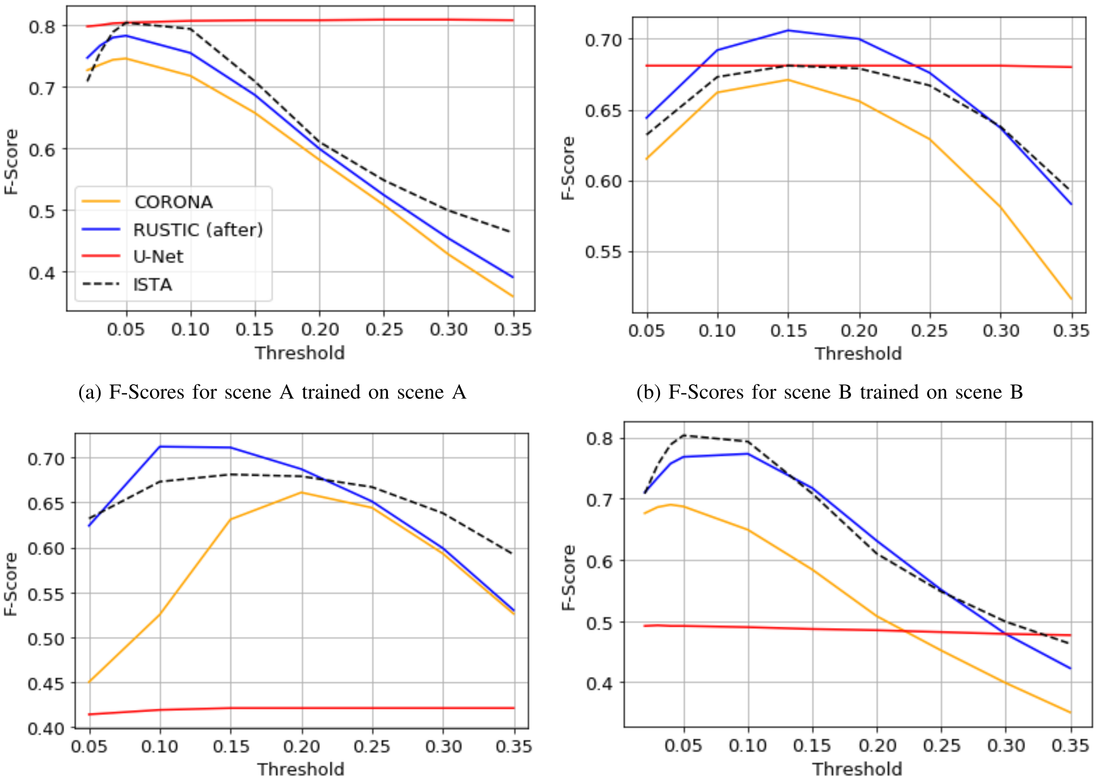
We experiment with three scenes, labeled A, B, and C, that are all 30 frames long. Only three scenes are used in this work because of the limited amount of synchronized camera/radar data available. We believe this amount still provides a sufficient demonstration because these three scenes have distinct background and foreground, contain varying amounts of undesirable foreground like shadows and reflections, and are dissimilar enough to test each model's propensity to overfit. We do not train the model on scene C because the ISTA results are quite poor as shown in Fig. 7.

For a fair comparison to the unrolled networks, the U-Net model is trained on a single sequence where each image is its own channel. Thus, the input to the U-Net model is of shape $(30, H, W)$. This allows the U-Net to leverage information from the entire sequence instead of single images to predict its output. We also use the generated ISTA targets to train the U-Net; however, it is important to note that the U-Net outputs represent the probability of each pixel being foreground, as is common practice in deep learning BFS models [3]. This means the U-Net does not perform the same low-rank+sparse separation as the unrolled networks and only predicts the presence of foreground. To generate the U-Net targets, we threshold the magnitude of the sparse components from ISTA to create one-hot probability distribution targets at each pixel. Thus, the U-Net and unrolled models work with the same training data up to this small thresholding modification to train the U-Net. We empirically choose 0.075 for scene A and 0.15 for scene B as the thresholds for $|\mathbf{S}_{mhw}| \in [0, 1]$.

B. Complexity

Like ISTA, the time complexity of each layer in RUSTIC and CORONA is dominated by the SVD and matrix multiplication operations in the SVT operation to update the low rank component. Thus, the time complexity for a k-layer network is $\mathcal{O}\left(k[W^2H^2M + WHM^2 + M^3]\right)$. The memory requirement is also substantial due to the SVD operation because matrices of size $H^2W^2 \times H^2W^2$ and $H^2W^2 \times M$ need to be stored during each forward pass through each layer. To make such computation tractable, we process the input image sequence in patches smaller than the image size (H, W). During training, each batch consists of one randomly selected patch. Then, for test-time inference we choose a stride length in each dimension less than or equal to the patch size and iterate over the entire image. For cases when the stride length is less than the patch size in either dimension, we take the mean of the regions with overlap.

[2] https://github.com/corey-snyder/radar-rgb-bfs



(a) F-Scores for scene A trained on scene A

(b) F-Scores for scene B trained on scene B

(c) F-Scores for scene B trained on scene A

(d) F-Scores for for scene A trained on scene B

Fig. 5. Comparison of RUSTIC, CORONA and U-Net models with their ISTA targets against various thresholds. The means from 5 trials are displayed. (a) and (c) are trained on scene A while (b) and (d) are trained on scene B.

C. Training Details

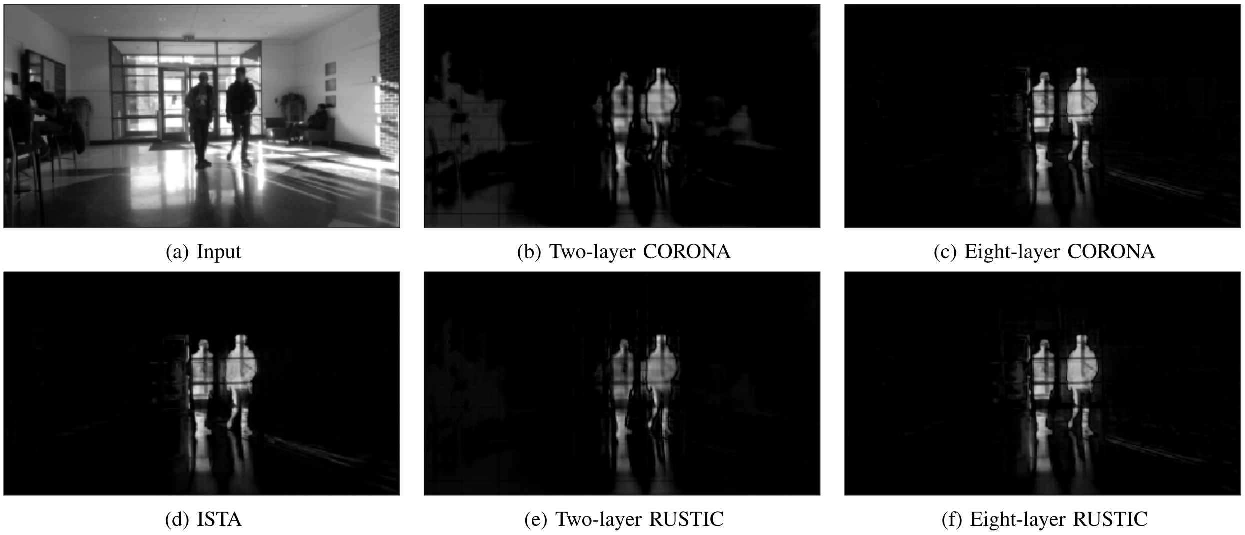
We train the models on a single sequence of 30 frames for 50,000 image patches. Our unrolled networks use 2D kernel sizes of 5 × 5 for the first 3 layers, 3 × 3 for all subsequent layers, and length 5 1D convolutions for radar data in all layers. We use the Adam optimizer [40] and a learning rate of 10^{-3} for the first 30,000 patches and 10^{-4} for the remaining 20,000. Furthermore, all models are run for five trials with five consistent random seeds shared across all architectures.

To address the high memory requirements of performing SVD, we use patch sizes of 80 × 80 for each input to the unrolled network. These image patches are sampled uniformly at random during training. To generate full image results for evaluation purposes, we tile the image into a grid of 80 × 80 patches with a stride length of 30 pixels in each dimension and average prediction results where there is overlap.

D. Results

For all figures and tables in this section, RUSTIC refers to the best performing configuration where the radar is used *after* the shrinkage operator unless noted otherwise. Quantitative results for two-layer unrolled models are presented in Fig. 5. We see a clear gap between the two unrolled networks in favor of the RUSTIC architecture. Notably, RUSTIC outperforms the ISTA targets when evaluated on scene B regardless of the scene it is trained on. Thus, RUSTIC provides real-time computation and superior performance on this scene. Figure 6 shows example sparse outputs for scene B from models trained on scene A. For the displayed image, only the two walking humans in the middle and the closing door are labeled as true foreground. From the figure, we see that the two-layer RUSTIC model detects the true foreground similarly to ISTA and CORONA but does a much better job of suppressing the shadows to the right of the humans. This suggests that the side-information from the radar successfully disagrees with the camera data and yields a more precise foreground. When we increase the network depth to eight layers, we see the results from RUSTIC and CORONA become more similar as the eight-layer RUSTIC model includes more shadows in its foreground. This phenomenon indicates that the radar is used less in deeper models. We also note that because no elevation data is collected by the radar, there are no cues to reduce the reflections below the humans. With access to elevation data, we would expect radar side-information to further alleviate this failure mode for camera data.

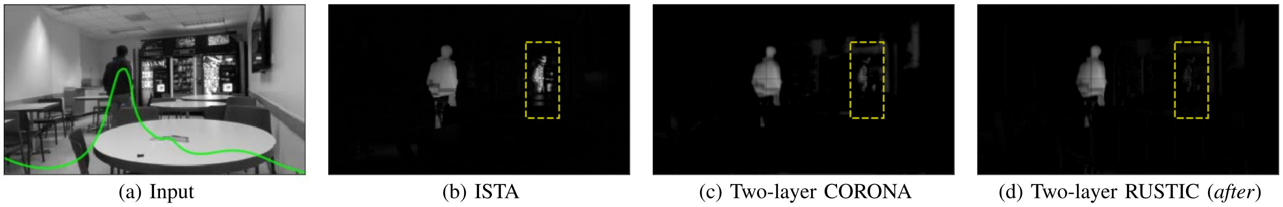
We also note in Fig. 5 the expected overfitting of the U-Net to its training data while poorly generalizing to unseen scenes. The U-Net is only able to match the maximum performance of the ISTA method in 5a and 5b because the U-Net targets are generated directly from thresholding the ISTA results. Lastly, as seen in Fig. 6e and corroborated by Fig. 5c and Table I, two-layer RUSTIC models perform the best qualitatively and according to F-score on scenes with high amounts of shadow. This is notable because such a shallow model (1) takes more influence from the radar and (2) is faster than its deeper



(a) Input (b) Two-layer CORONA (c) Eight-layer CORONA

(d) ISTA (e) Two-layer RUSTIC (f) Eight-layer RUSTIC

Fig. 6. Results from scene B for models trained on scene A. Images (b)-(f) contain the magnitudes of the sparse outputs. The RUSTIC models used in (e) and (f) incorporate the radar *after* the shrinkage operator. The two-layer RUSTIC model with radar does the best job of suppressing the shadows and static humans/furniture on the sides while performing just 0.57% slower than the fastest model (2-Layer CORONA). At eight layers, the models perform relatively similarly. Moreover, the radar data seems less influential in (f) than in (e) since there is a stronger presence of shadows on the right.



(a) Input (b) ISTA (c) Two-layer CORONA (d) Two-layer RUSTIC (*after*)

Fig. 7. The human pictured above in scene C stands in front of the right vending machine for the majority of the frames (not pictured in this frame) thus causing a ghost human to incorrectly appear in the foreground and background simultaneously as emphasized by the yellow rectangle. Here we see the RUSTIC model in (d) is the only model to properly suppress this error. For context, the radar data is superimposed on (a).

TABLE I

THE HIGHEST ACHIEVED F-SCORES FOR TWO AND EIGHT-LAYER MODELS OF CORONA AND RUSTIC. THE MEANS FROM 5 TRIALS ARE DISPLAYED. THE BEST RESULTS FOR EACH NETWORK DEPTH ARE BOLDED AND THE BEST RESULTS FOR EACH ROW ARE UNDERLINED

Scene (Train, Eval)	2 Layer CORONA [10]	2 Layer RUSTIC	8 Layer CORONA [10]	8 Layer RUSTIC
(A,A)	0.745	**0.782**	**0.797**	**0.797**
(A,B)	0.661	**0.712**	0.694	**0.702**
(B,A)	0.690	**0.773**	**0.808**	0.798
(B,B)	0.671	**0.706**	0.688	**0.691**

TABLE II

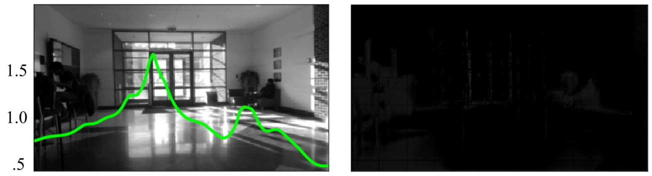
NUMBER OF TRAINABLE PARAMETERS AND AVERAGE INFERENCE TIME FOR EACH METHOD USING SETUP FROM SECTION V-A. ISTA IS RUN ON AN INTEL® CORE$^{\text{TM}}$ I7 7TH GEN PROCESSOR WHILE THE NETWORKS ARE RUN ON AN NVIDIA GTX 1070 GPU

Method	# of Parameters	Inference Time (s)	Mean FPS
ISTA/RISTA [13]	0	20.612 ± 0.099	1.46
2-Layer CORONA [10]	316	1.157 ± 0.020	25.93
2-Layer RUSTIC	328	1.150 ± 0.002	26.08
8-Layer CORONA [10]	784	2.350 ± 0.005	12.76
8-Layer RUSTIC	832	2.368 ± 0.014	12.67
U-Net [12]	13,412,766	0.399 ± 0.003	75.23

counterparts. We argue for this first point in particular since the unrolled networks with and without radar perform closely with the deeper eight-layer architecture. Supporting numerical results for both two and eight-layer models are presented in Table I. For the following comparisons between RUSTIC and CORONA, we will use models with two layers.

We also compare the number of parameters and computation time of each method in Table II. We see the expected dramatic gap between the unrolled models and the standard U-Net in number of parameters. Lastly, we observe that both unrolled models support real-time computation unlike their iterative ISTA counterpart.

1) Sleeping Foreground: In scene C, we see dramatic qualitative results as we address a situation with *sleeping foreground*. Sleeping foreground refers to the scenario where a foreground object, in this case a human, behaves as clear moving foreground for some frames and then remains relatively still for a large portion of the remaining frames. For the sequence depicted in Fig. 7, the human stands in front of the vending machine on the right for the last 18/30 frames causing both the ISTA algorithm and the two-layer CORONA network to mistakenly absorb them into the low-rank background. As a result, when the person isn't standing at the vending machine



(a) Input image with radar return (b) RUSTIC sparse output

Fig. 8. In (a), we see a camera image with no visible foreground while the radar return after clutter suppression is overlayed on top. Because there is no visible foreground, (b) should be entirely zero (black), which is nearly the case. For context, the walking humans earlier in the frame had returns with magnitudes between 1.00 and 2.75.

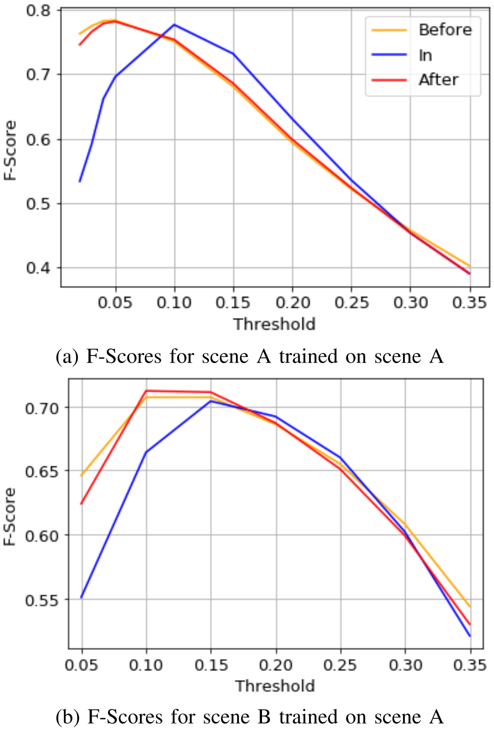
like the frame shown in the figure, the sparse foreground must compensate by outputting a *ghost human* in the foreground component. Yet, for the same image, we see that RUSTIC is able to much more effectively suppress the appearance of this ghost human. As mentioned earlier, because deeper networks rely more on the camera data and less on the radar data, the eight-layer models both with and without radar perform poorly and are unable to suppress this instance of incorrect foreground.

2) Dealing With Radar False Positives: Many instances of undesirable foreground in the camera data can be thought of as false positives that are suppressed by the incorporation of the radar data. As mentioned in Fig. 1, the opposite may also occur when the radar mistakenly detects motion when there is none to be seen in the corresponding image. This could be due multipath or motion that is occluded to the camera i.e. behind a wall or inside an opaque container. In one instance, shown in Fig. 8, there are two visible peaks in the radar return after clutter suppression as shown overlayed on the image. For context, the walking humans in this sequence have radar returns with magnitudes ranging between 1.00-2.75. Because there actually is no visible foreground, the sparse component in 8b should be entirely zero (black). Despite this misleading radar return, RUSTIC correctly suppresses the foreground component thus demonstrating the model's ability to suppress false positives from either sensing modality.

3) Comparison of the Models With Radar: Thus far, all results generated using RUSTIC incorporate the radar *after* the shrinkage operator. In Fig. 9, we offer a comparison of the three different models described in (15), (16), (17) and depicted in Fig. 3. All three models were trained on scene A and tested on scene B.

In Fig. 9a, we see nearly identical performance for the *before* and *after* models. We also see a relatively high peak in training performance for the *in* model but at a higher threshold. This suggests the *in* model produces sparse foregrounds with lower precision.

Furthermore, during our experimentation, we noticed that certain training runs for the *in* model resulted in models that incorrectly predict the sparse components as all zeros. Thus, the results in Fig. 9 only include the models that do not suffer from this instability. We address this issue during training in the following subsection.



(a) F-Scores for scene A trained on scene A

(b) F-Scores for scene B trained on scene A

Fig. 9. A comparison of the *in*, *before*, and *after* two-layer RUSTIC models. The means from 5 trials are displayed.

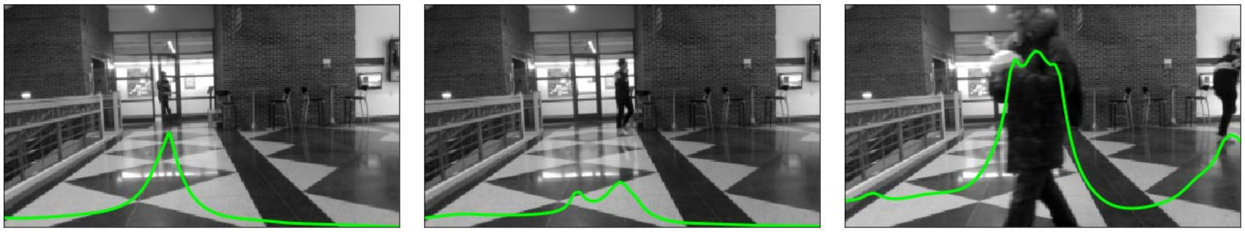
4) Ablation Study With Cosine Similarity Loss: As mentioned above, the models that incorporate the radar *in* the shrinkage operator are prone to local minima where the low rank outputs are learned correctly and the sparse components give all zeros. To rectify this, we add to the loss function a scaled cosine similarity term between the l_1 norm of the columns in $\mathbf{S}_m$ and $\mathbf{R}_m$

$$\alpha \frac{\langle \sum_{h=1}^{H} |\mathbf{S}_{mh}|, \mathbf{R}_m \rangle}{\left\| \sum_{h=1}^{H} |\mathbf{S}_{mh}| \right\|_2 \|\mathbf{R}_m\|_2} \tag{19}$$

where $\alpha \in [0, 1]$. This loss term assumes the amount of sparse foreground in a given column is proportional to its radar return. The absolute value ensures that negative and positive foreground intensities are treated identically. In our experiments, we empirically set $\alpha = 10^{-3}$ to appropriately balance the MSE loss. With this choice, we observe that all runs avoid any local minima and the performance is otherwise unaffected for better or worse.

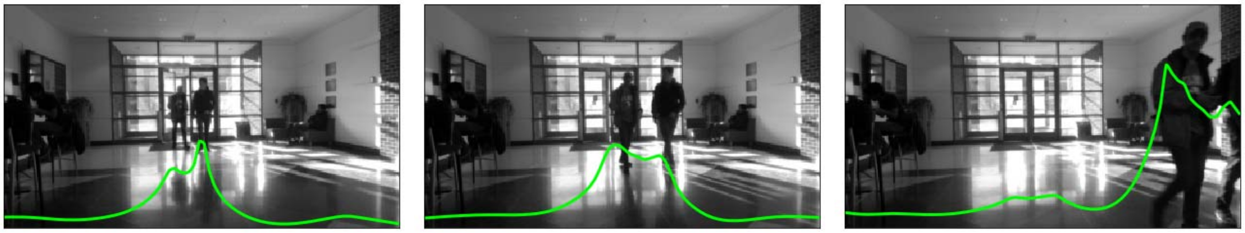
VI. Conclusion

In this work, we present a number of contributions to BFS. First, we motivated the incorporation of radar data into the RPCA objective and introduced an associated iterative solver called RISTA. We then unrolled our iterative algorithm into our RUSTIC model and tested our approach in the unsupervised setting where no ground-truth is available. We found that RUSTIC provided real-time computation without sacrificing the performance from the associated iterative solver. While we do notice some convergence issues with incorporating the



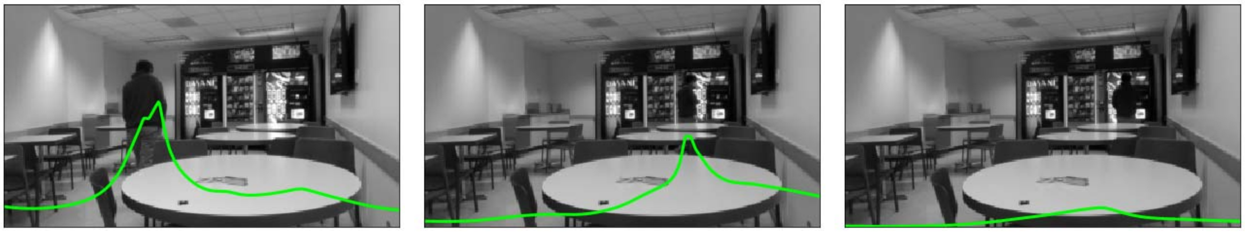
(a) Frame 0 (b) Frame 10 (c) Frame 20

Fig. 10. Sample images and radar data from scene A.



(a) Frame 0 (b) Frame 10 (c) Frame 20

Fig. 11. Sample images and radar data from scene B.



(a) Frame 0 (b) Frame 10 (c) Frame 20

Fig. 12. Sample images and radar data from scene C.

radar *in* the shrinkage operator, we mitigated this issue with the addition of a cosine similarity loss term during training.

We also demonstrated strong performance in scenarios when the camera data and radar disagree. We showed that the two-layer RUSTIC network is able to effectively suppress shadows and ignore sleeping foreground objects. Moreover, in the case with improper radar returns, we saw that the sparse output did not contain strong unwanted foreground when the radar incorrectly encouraged otherwise.

Finally, we saw in Fig. 6 that deeper models performed more closely to ISTA and seemed to incorporate radar information less. This phenomenon was most pronounced for quantitative results with two-layer unrolled networks as RUSTIC clearly outperformed CORONA. This provides evidence that deeper unrolled models may not always be best, especially when additional modalities are available.

While this work does demonstrate the efficacy of using radar reflections at a given bearing for BFS, much of the radar information remains unused. For example, with priors on the types of targets that may be observed in a scene, the radar's range and magnitude information could provide insight on how much area in pixels the targets might occupy. Furthermore, additional processing on the velocity information could also prove useful in extracting desirable foreground. This velocity information could be incorporated into a tracking scheme that yields more reliable and consistent foreground concepts. Moreover, in some cases users may only be interested in viewing foreground targets that fall within a certain range of Doppler velocities. This might be useful in distinguishing between moving vehicles and walking pedestrians.

Finally, we believe that the sensor fusion methods presented in this work are not limited to radar. Other sensors such as sonar and lidar likely could also be used as long as the processing can eliminate static clutter reliably.

Appendix
Sample Images

See Figs. 10–12.

References

[1] B. Garcia-Garcia, T. Bouwmans, and A. J. R. Silva, "Background subtraction in real applications: Challenges, current models and future directions," *Comput. Sci. Rev.*, vol. 35, Feb. 2020, Art. no. 100204. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574013718303101

[2] T. Bouwmans and E. H. Zahzah, "Robust PCA via principal component pursuit: A review for a comparative evaluation in video surveillance," *Comput. Vis. Image Understand.*, vol. 122, pp. 22–34, May 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1077314213002294

[3] T. Bouwmans, S. Javed, M. Sultana, and S. K. Jung, "Deep neural network concepts for background subtraction: A systematic review and comparative evaluation," *Neural Netw.*, vol. 117, pp. 8–66, Sep. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608019301303

[4] T. Bouwmans, "Traditional and recent approaches in background modeling for foreground detection: An overview," *Comput. Sci. Rev.*, vols. 11–12, pp. 31–66, May 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574013714000033

[5] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 399–406.

[6] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Process. Mag.*, vol. 38, no. 2, pp. 18–44, Mar. 2021.

[7] Y. Li, M. Tofighi, J. Geng, V. Monga, and Y. C. Eldar, "Efficient and interpretable deep blind image deblurring via algorithm unrolling," *IEEE Trans. Comput. Imag.*, vol. 6, pp. 666–681, 2020.

[8] R. Hyder, Z. Cai, and M. S. Asif, "Solving phase retrieval with a learned reference," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 425–441.

[9] N. Shlezinger, N. Farsad, Y. C. Eldar, and A. J. Goldsmith, "ViterbiNet: A deep learning based Viterbi algorithm for symbol detection," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3319–3331, May 2020.

[10] O. Solomon *et al.*, "Deep unfolded robust PCA with application to clutter suppression in ultrasound," *IEEE Trans. Med. Imag.*, vol. 39, no. 4, pp. 1051–1063, Apr. 2020.

[11] T.-Y. Lim, S. A. Markowitz, and M. N. Do, "RaDICaL: A synchronized FMCW radar, depth, IMU and RGB camera data dataset with low-level FMCW radar signals," *IEEE J. Sel. Topics Signal Process.*, vol. 15, no. 4, pp. 941–953, Jun. 2021.

[12] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2015, pp. 234–241.

[13] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 1, pp. 1–37, 2011.

[14] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen, and Y. Ma, "Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix," Coordinated Sci. Lab., Urbana, IL, USA, Tech. Rep. UILU-ENG-09-2214, DC-246, 2009.

[15] C. Qiu and N. Vaswani, "Real-time robust principal components' pursuit," in *Proc. 48th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep. 2010, pp. 591–598.

[16] J. Feng, H. Xu, and S. Yan, "Online robust PCA via stochastic optimization," in *Advances in Neural Information Processing Systems*, vol. 26, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, 2013.

[17] N. Vaswani, T. Bouwmans, S. Javed, and P. Narayanamurthy, "Robust subspace learning: Robust PCA, robust subspace tracking, and robust subspace recovery," *IEEE Signal Process. Mag.*, vol. 35, no. 4, pp. 32–55, Jul. 2018.

[18] P. Rodriguez and B. Wohlberg, "Incremental principal component pursuit for video background modeling," *J. Math. Imag. Vis.*, vol. 55, no. 1, pp. 1–18, 2016.

[19] P. Narayanamurthy and N. Vaswani, "A fast and memory-efficient algorithm for robust PCA (MEROP)," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 4684–4688.

[20] X. Guo, X. Wang, L. Yang, X. Cao, and Y. Ma, "Robust foreground detection using smoothness and arbitrariness constraints," in *Proc. Eur. Conf. Comput. Vis.* New York, NY, USA: Springer, 2014, pp. 535–550.

[21] S. Javed, S. Ki Jung, A. Mahmood, and T. Bouwmans, "Motion-aware graph regularized RPCA for background modeling of complex scenes," in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, Dec. 2016, pp. 120–125.

[22] W. Xu, T. Xia, and C. Jing, "Background modeling from video sequences via online motion-aware RPCA," *Comput. Sci. Inf. Syst.*, vol. 18, no. 4, pp. 1411–1426, 2021.

[23] S. Javed, T. Bouwmans, M. Sultana, and S. K. Jung, "Moving object detection on RGB-D videos using graph regularized spatiotemporal RPCA," in *Proc. Int. Conf. Image Anal. Process.* New York, NY, USA: Springer, 2017, pp. 230–241.

[24] H. Fu and H. Liu, "Online RPCA background modeling based on color and depth data," in *Proc. Chin. Intell. Syst. Conf.* New York, NY, USA: Springer, 2019, pp. 511–517.

[25] M. Mandal and S. K. Vipparthi, "An empirical review of deep learning frameworks for change detection: Model design, experimental frameworks, challenges and research needs," *IEEE Trans. Intell. Transp. Syst.*, early access, May 19, 2021, doi: 10.1109/TITS.2021.3077883.

[26] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "CDNet 2014: An expanded change detection benchmark dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2014, pp. 393–400.

[27] L. Maddalena and A. Petrosino, "Towards benchmarking scene background initialization," in *New Trends in Image Analysis and Processing—ICIAP 2015 Workshops*, V. Murino, E. Puppo, D. Sona, M. Cristani, and C. Sansone, Eds. Genova, Italy, 2015.

[28] L. A. Lim and H. Y. Keles, "Learning multi-scale features for foreground segmentation," *Pattern Anal. Appl.*, vol. 23, no. 3, pp. 1369–1380, 2020, doi: 10.1007/s10044-019-00845-9.

[29] Y. Wang, Z. Luo, and P.-M. Jodoin, "Interactive deep learning method for segmenting moving objects," *Pattern Recognit. Lett.*, vol. 96, pp. 66–75, Jan. 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167865516302471

[30] J. Zhang, X. Zhang, Y. Zhang, Y. Duan, Y. Li, and Z. Pan, "Meta-knowledge learning and domain adaptation for unseen background subtraction," *IEEE Trans. Image Process.*, vol. 30, pp. 9058–9068, 2021.

[31] M. O. Tezcan, P. Ishwar, and J. Konrad, "BSUV-Net: A fully-convolutional neural network for background subtraction of unseen videos," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 2774–2783.

[32] J. H. Giraldo, S. Javed, N. Werghi, and T. Bouwmans, "Graph CNN for moving object detection in complex environments from unseen videos," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 225–233.

[33] B. Hou, Y. Liu, N. Ling, L. Liu, and Y. Ren, "A fast lightweight 3D separable convolutional neural network with multi-input multi-output for moving object detection," *IEEE Access*, vol. 9, pp. 148433–148448, 2021.

[34] H. Fu, Z. Ma, B. Zhao, Z. Yang, Y. Jiang, and M. Zhu, "Lightweight convolutional neural network for foreground segmentation," in *Proceedings of 2021 Chinese Intelligent Systems Conference*, Y. Jia, W. Zhang, Y. Fu, Z. Yu, and S. Zheng, Eds. Singapore: Springer, 2022, pp. 811–819.

[35] T.-Y. Lim *et al.*, "Radar and camera early fusion for vehicle detection in advanced driver assistance systems," in *Proc. Mach. Learn. Auton. Driving Workshop 33rd Conf. Neural Inf. Process. Syst.*, vol. 2, 2019, pp. 1–11.

[36] M. Sultana, A. Mahmood, S. Javed, and S. K. Jung, "Unsupervised deep context prediction for background estimation and foreground segmentation," *Mach. Vis. Appl.*, vol. 30, no. 3, pp. 375–395, Apr. 2019, doi: 10.1007/s00138-018-0993-0.

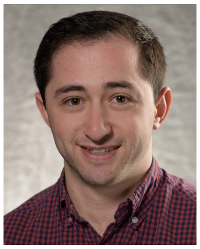
[37] M. O. Tezcan, P. Ishwar, and J. Konrad, "BSUV-Net 2.0: Spatio-temporal data augmentations for video-agnostic supervised background subtraction," *IEEE Access*, vol. 9, pp. 53849–53860, 2021.

[38] J. Capon, "High-resolution frequency-wavenumber spectrum analysis," *Proc. IEEE*, vol. 57, no. 8, pp. 1408–1418, Aug. 1969.

[39] P. Sprechmann, A. M. Bronstein, and G. Sapiro, "Learning efficient sparse and low rank models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1821–1833, Sep. 2015.

[40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, Y. Bengio and Y. LeCun, Eds., San Diego, CA, USA, May 2015, pp. 1–15.

Spencer Markowitz received the B.S. degree in electrical engineering from the University of Illinois at Urbana-Champaign, where he is a graduate student with the Electrical and Computer Engineering Department. His primary research interests include FMCW radar, computer vision, object tracking, and deep learning.



Corey Snyder received the B.S. degree in electrical engineering and the M.S. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign in 2018 and 2020, respectively, where he is a graduate student with the Electrical and Computer Engineering Department. His research interest includes semi-supervised, weakly supervised, and unsupervised learning for computer vision.

Yonina C. Eldar (Fellow, IEEE) received the B.Sc. degrees in physics and electrical engineering from Tel Aviv University (TAU), Tel Aviv-Yafo, Israel, in 1995 and 1996, respectively, and the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, in 2002.

She is currently a Professor with the Department of Mathematics and Computer Science, Weizmann Institute of Science, Rehovot, Israel. She was previously a Professor with the Department of Electrical Engineering, Technion, where she held the Edwards Chair in engineering. She is also a Visiting Professor with MIT, a Visiting Scientist with the Broad Institute, and an Adjunct Professor with Duke University; and was a Visiting Professor with Stanford University. She is the author of the book *Sampling Theory: Beyond Bandlimited Systems* and the co-author of four other books published by Cambridge University Press. Her research interests are in the broad areas of statistical signal processing; sampling theory and compressed sensing; learning and optimization methods; and their applications to biology, medical imaging, and optics.

Dr. Eldar was a member of the Young Israel Academy of Science and Humanities and the Israel Committee for Higher Education. She is a member of the Israel Academy of Sciences and Humanities (elected 2017) and a fellow of EURASIP. She is also a member of the IEEE Sensor Array and Multichannel Technical Committee and serves on several other IEEE committees. In the past, she was a Horev Fellow of the Leaders in Science and Technology Program at the Technion and a fellow of Alon. She was also a Distinguished Lecturer of the Signal Processing Society and a member of the IEEE Signal Processing Theory and Methods and Bio Imaging Signal Processing technical committees. She has received many awards for excellence in research and teaching, including the IEEE Signal Processing Society Technical Achievement Award in 2013, the IEEE/AESS Fred Nathanson Memorial Radar Award in 2014, and the IEEE Kiyo Tomiyasu Award in 2016. She has received the Michael Bruno Memorial Award from the Rothschild Foundation, the Weizmann Prize for Exact Sciences, the Wolf Foundation Krill Prize for Excellence in Scientific Research, the Henry Taub Prize for Excellence in Research (twice), the Hershel Rich Innovation Award (three times), the Award for Women with Distinguished Contributions, the Andre and Bella Meyer Lectureship, the Career Development Chair at the Technion, the Muriel and David Jacknow Award for Excellence in Teaching, and the Technion's Award for Excellence in Teaching (two times). She received several best paper awards and best demo awards together with her research students and colleagues, including the SIAM Outstanding Paper Prize, the UFFC Outstanding Paper Award, the Signal Processing Society Best Paper Award, and the *IET Circuits, Devices and Systems* Premium Award; and was selected as one of the 50 most influential women in Israel and Asia, and is a highly cited researcher. She was the co-chair and the technical co-chair of several international conferences and workshops. She served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING, the *EURASIP Journal on Advances in Signal Processing*, the *SIAM Journal on Matrix Analysis and Applications*, and the *SIAM Journal on Imaging Sciences*. She is the Editor-in-Chief of *Foundations and Trends in Signal Processing*.

Minh N. Do (Fellow, IEEE) was born in Vietnam in 1974. He received the B.Eng. degree in computer engineering from the University of Canberra, Australia, in 1997, and the Dr.Sci. degree in communication systems from the Swiss Federal Institute of Technology Lausanne (EPFL), Switzerland, in 2001.

Since 2002, he has been on the Faculty of the University of Illinois at Urbana-Champaign (UIUC), where he is currently the Thomas and Margaret Huang Endowed Professor in signal processing and data science with the Department of Electrical and Computer Engineering, and holds affiliate appointments with the Coordinated Science Laboratory, Department of Bioengineering, and the Department of Computer Science, Beckman Institute for Advanced Science and Technology. From 2020 to 2021, he is on leave from UIUC to serve as the Vice Provost for VinUniversity, Vietnam.

Dr. Do was a member of several IEEE technical committees on signal processing. He was elected as a fellow of IEEE in 2014 for his contributions to image representation and computational imaging. He received the Silver Medal from the 32nd International Mathematical Olympiad in 1991, the University Medal from the University of Canberra in 1997, the Doctorate Award from the EPFL in 2001, the CAREER Award from the National Science Foundation in 2003, the Xerox Award for Faculty Research from UIUC in 2007, and the Young Author Best Paper Award from IEEE in 2008. He has contributed to several tech-transfer efforts, including as the Co-Founder and the CTO of Personify, and the Chief Scientist of Misfit. He was an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING.