Filtering Synthetic Aperture Radar Images with Haar Wavelets and Principal Components

Sara L. Daley

3/19/2020

Abstract

The goal of this paper is to filter Synthetic Aperture Radar (SAR) using two different methods. The first method being Haar Wavelet Transforms, and the second method being Principal Component Analysis (PCA). We consider two different reasons to do so: which is easier as a human to see, and for putting the filtered image into a deep learning model channel for future work on image classification.

1 Introduction and Overview

Radar imaging has been around for many years now. It is a topic that is extensively studied, and continues to be studied. There are many techniques done to Synthetic Aperture Radar (SAR) images to make them more human readable, such as Taylor Weighting to remove sidelobe artifacts. However, with emerging fields such as Data Science, we not only revisit techniques to view how they compare to the human eye, but also view them for their ability to feed into a deep learning model channel.[1]

1.1 Synthetic Aperture Radar

SAR can take high quality images of the Earth's surface from either an aircraft or spacecraft. SAR has the advantage to see through most clouds, weather, and is not limited to imaging only during the daytime[1].

[1]This paper does not seek to put the results through a channel at this time due to resource availability (SAR images are large and computationally intensive to work with).

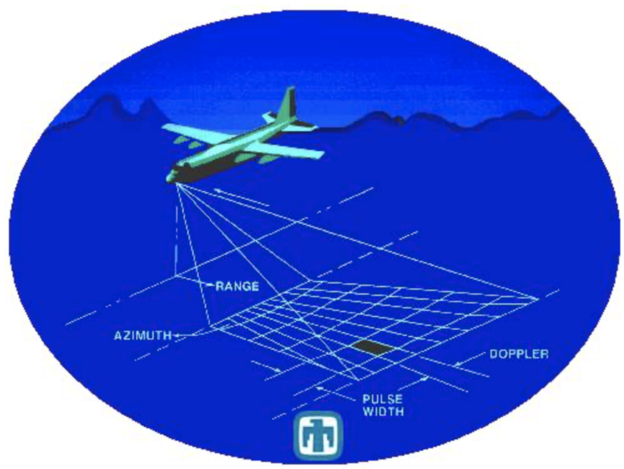

Figure 1: Illustration of Airborne SAR from Sandia National Laboratories[2]

As the aircraft or spacecraft moves along its flight path, it will pulse out microwaves which reach down to the surface, reflect back, and is then collected for analysis. Depending on the object, the microwaves will either be absorbed or reflected. Consider Figure 1, which shows the basics in image creation. Range is the distance from the radar to the target. Azimuth is perpendicular to the range, and its resolution is dependent on the size of the antenna as it focuses the energy (transmitted and received) into a beam.

Let's now shift our focus to consider noise in a SAR image. A typical target response will have intensity such as Figure 2. The closer the sidelobe intensity is to the mainlobe, the image will have bright streaks around an object[3], as in Figure 3. One industry standard to remove sidelobe artifacts in the image is to apply Taylor Weighting which brings down the sidelobes in intensity. The resolution of the needed image will depend on the image use. In essence, target classification will require a higher resolution than terrain mapping. There are many other aspects of SAR not considered here as they are beyond the scope of this paper.

1.2 Dataset

The data used for this experiment are SAR images based in Sandia National Laboratories' Ground-based SAR Applications Testbed File Format (GFF). Within their website, they have a few images available for download as well

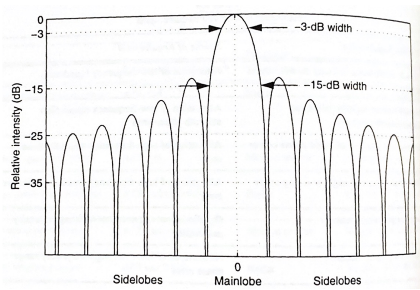

Figure 2: Illustration of one dimensional intensity impulse response with uniform weighting[3].

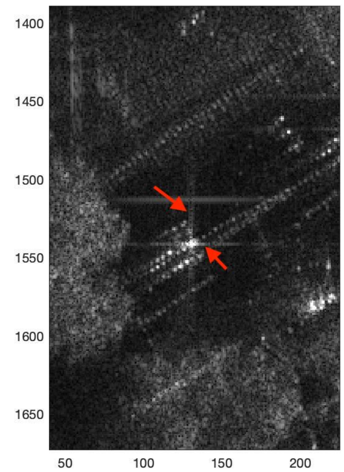

Figure 3: An example of sidelobe artifacts in an image.

as starter code for viewing the images.

2 Theoretical Background

The goal in filtering an image is to reduce noise without dulling out or removing key features; however, a balance between noise and features must be found. A high degree of filtering blurs, thus key features aren't retained[4].

2.1 Haar Wavelet

Wavelets are a way to represent the signal (in this case, the image) as an expansion basis. Instead of translating a filter, such as Gaussian, across the signal; the filter is also scaled. The Haar wavelet is given by the function,

$$\psi(x) = \begin{cases} 1 & 0 \leq x < 1/2 \\ -1 & 1/2 \leq x < 1 \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

This wavelet is a good candidate for describing localized signals in space[4]. In Equation 1, the translation is zero and scaling is unity. In order to reconstruct the denoised image, the image is decomposed into $\psi_{m,n}(x)$ where m is the scaling parameter and n is translation[4]. An example of how different scaling

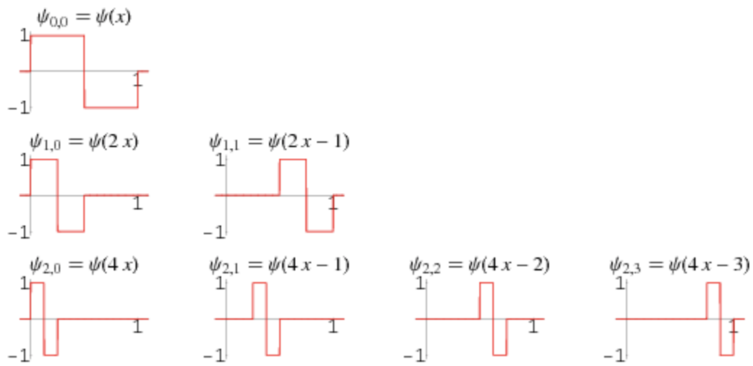

Figure 4: Visualization of Haar Wavelet with various scaling and translations. Image from MathWorld[5].

and translations can change the shape of the wavelet are shown in Figure 4. When using wavelets for denoising an image, at each level of decomposition the filter is applied. This will remove the high frequency components at the level of decomposition chosen. Matlab recommends choosing the minimum of $floor(log_2 N)$ (where N is the number of samples in the signal) as the level of decomposition.

2.2 Principal Component Analysis

For PCA, the main point to refer to is basis transformations. If you start with a vector $\mathbf{x}$ and scale it by $\mathbf{A}$ the result is a new magnitude and direction. A singular value decomposition (SVD) is a factorization of the image (represented as a matrix) into a number of constitutive components all of which have a specific meaning in applications[4]. The full SVD decomposition takes the form[4]

$$\mathbf{A} = \mathbf{U\Sigma V}^* \tag{2}$$

where

$$\mathbf{U} \in \mathbb{C}^{\mathbf{mxm}} \text{ is unitary} \tag{3}$$

$$\mathbf{V} \in \mathbb{C}^{\mathbf{nxn}} \text{ is unitary} \tag{4}$$

$$\mathbf{\Sigma} \in \mathbb{R}^{\mathbf{mxn}} \text{ is diagonal} \tag{5}$$

This is saying that you can decompose a matrix that you don't know anything about (**A** here) and find out information from its displacement (**U**), singular values ($\mathbf{\Sigma}$)), and direction (**V**). The theorem behind the SVD:

Theorem 1. *Every matrix* $\mathbf{A} \in \mathbb{C}^{\mathbf{mxn}}$ *has a singular value decomposition (Equation 2). Furthermore, the singular values* σ_j *are uniquely determined, and, if* **A** *is square and the* σ_j *distinct, the singular vectors* $\mathbf{u_j}$ *and* $\mathbf{v_j}$ *are uniquely determined up to complex signs (complex scalar factors of absolute value 1).[4]*

According to Kutz, "the SVD gives a type of least-square fitting algorithm, allowing us to project the matrix onto low dimensional representations in a formal, algorithmic way."[4] Principal Component Analysis is a variant of Singular Value Decomposition. We want to look at the principal components; therefore we want to focus on the variance of the data.

The SVD produces principal components, and the modes are produced by the L^2 norm. This can lead to issues when the data has noise, as the singular values and modes will not be able to accurately describe the data[4]. To limit the negative impact the L^1 norm is used, and the matrix decomposition becomes

$$\mathbf{A} = \mathbf{L} + \mathbf{S} + \mathbf{N} \tag{6}$$

where **L** is the low-rank, **S** is the sparse data, and **N** is the (small) error which represents the dense, small noise[4].

3 Computational Results

In determining a "good" filter, there are many avenues to consider, as which filter to choose will depend on the problem one is trying to solve. For example, for the human eye, a good filter will remove the sidelobe artifacts as it produces bright streaks. Sidelobes can hide the edges of the object or make one object look like two. For a machine it may be helpful to remove as much of the undesired background as possible making the desired targets more visible. Note that the images shown in this section do not have the $x - y$ axes labeled. That is because the axes are pixel coordinates. Pixels are not easy to distinguish in a small image, thus it was chosen to draw your attention to the areas of focus using colored arrows, lines or boxes.

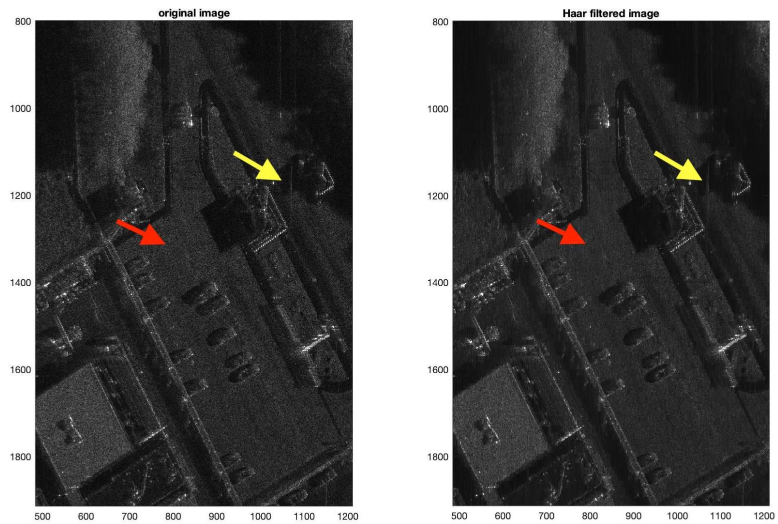

Figure 5: The unfiltered image is on the left. The Haar filtered image is on the right. We notice a few slightly subtle changes between the two.

3.1 Filtering with the Haar Wavelet

To denoise the image using the Haar Wavelet, the function `wdenoise` was used. Consider Figure 5 for the first scene, where we compare both the

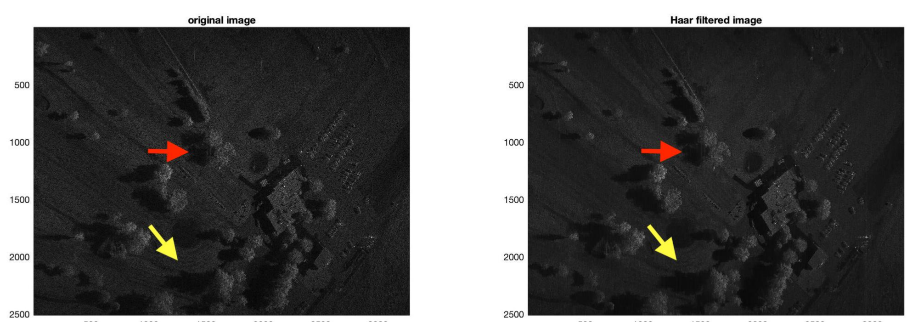

Figure 6: The unfiltered image is on the left. The Haar filtered image is on the right. We notice a few slightly subtle changes between the two, but overall a smoother image on the right.

unfiltered image, and the filtered image using a Haar wavelet. When looking at the red arrow, it is noticed that the speckling of what might be blacktop is more smoothed in the filtered image on the right. When looking at the

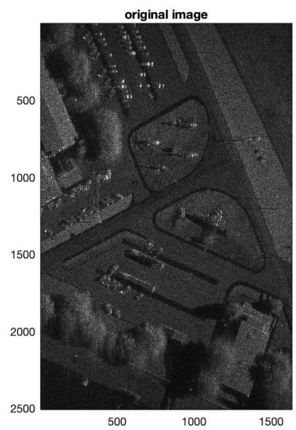

Figure 7: An unfiltered image scene that contains vehicles, helicopters, and an aircraft.

yellow arrow, it is noticed that the streaks from the sidelobe artifacts have not lessened.

Focusing now on a different scene; in Figure 6, it is noticed that the image as a whole is smoother. When looking at the yellow arrow, it may be noticed that the feature of the pattern in the grass is diminishing in the image on the right. The red arrow shows that with those two trees, the shadow is darker (less speckled), but the trees themselves are blurred more on the right.

The last scene to consider for Haar wavelets is shown in its original form in Figure 7, and is full of interesting items. There are vehicles, helicopters, an aircraft, and scenery. In Figure 8 it is attempted to zoom in on the areas of interest. In looking at the difference between the vehicles, the filtered image does not help in distinguishing if that bright spot is multiple vehicles. It also does not help in removing the sidelobe artifacts. In looking at the difference between helicopters it is noticed that at the nose, the filtered image is almost worst as it adds vertical streaks. In looking at the aircraft, there is no noticeable feature difference.

3.2 Filtering with Robust PCA

To denoise the image using the `robust_alm_rpca`, the positive weighting parameter, λ, was set to $\lambda = 0.2$. Focusing on Figure 9 for the first scene

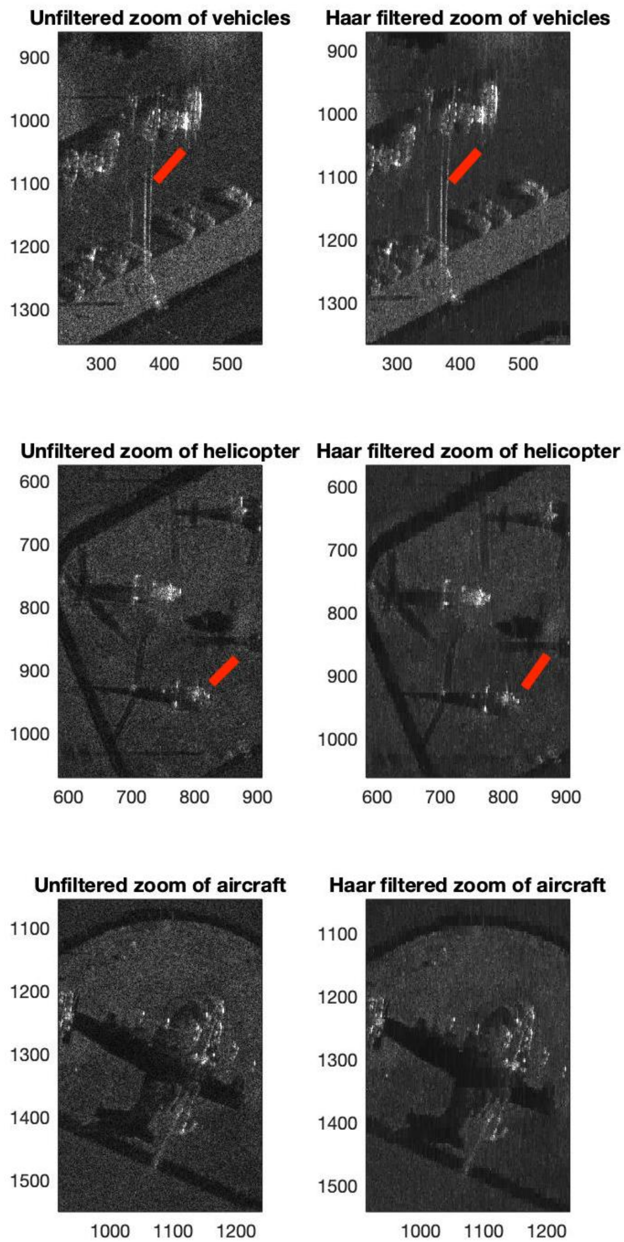

Figure 8: Comparing unfiltered and Haar filtered images for vehicles, helicopters, and an aircraft.

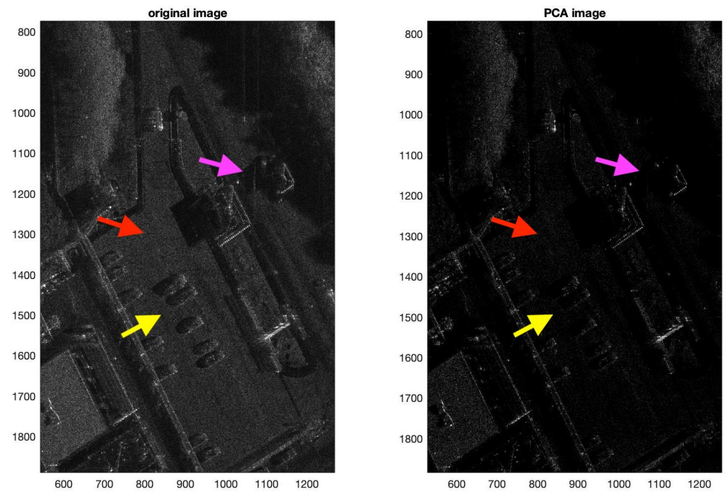

Figure 9: The unfiltered image is on the left. The PCA image is on the right.

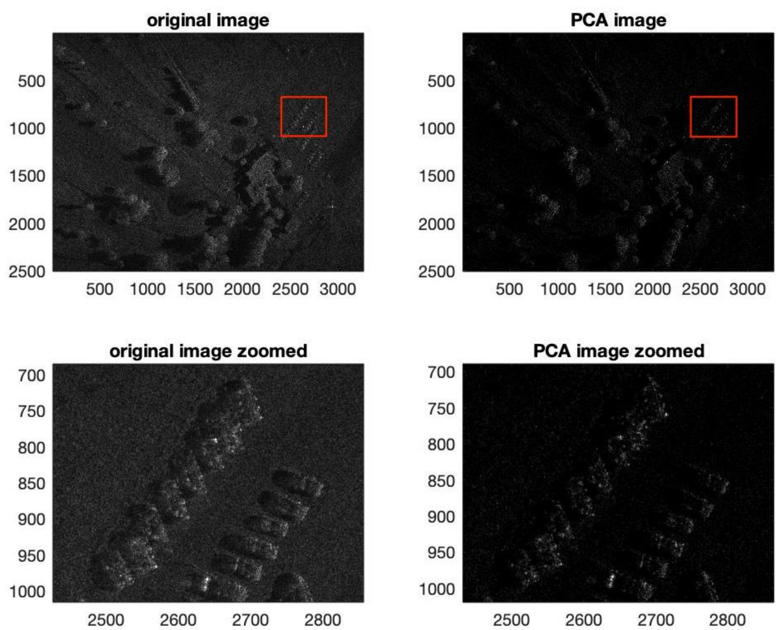

Figure 10: The unfiltered image is on the left. The PCA image is on the right. The bottom images are zoomed in areas of the red box in the upper images.

considered, there are more distinctions happening in the PCA image than for the Haar wavelet. Right away it is noticed when looking at the red arrow that the background/ground is darker. With the yellow arrow, the two vehicles are still visible- but just so. Mostly what was retained with the vehicles were the edges. This could be because the vehicles are similar in color as the background and the edges are the brightest. When looking at the magenta arrow, the sidelobe artifacts that were made more prominent in Figure 5 have almost completely been removed.

Consider now Figure 10 for the next scene. The top two images are similar to Figure 6, where the original is on the left, filtered on the right. The red boxes show where the image was zoomed to create the two images below. The ground has successfully been removed, and more of the vehicle shape is retained than in Figure 9; however, for a human (at least) the vehicles are less distinguishable.

Focus for a brief second back on Figure 7, as it is the scene considered for the last set of images, Figure 11. As in Figure 8, the sidelobe artifacts did not disappear, although with PCA they did lessen. The vehicles look slightly more distinct; however, the vehicles below the red line are not as noticeable. With the helicopter, it does make the cabin brighter as it is now against a dark background. Unfortunately, it is no longer visible as a helicopter. With the aircraft it does stand out, but the signal return can appear as multiple targets if the size of the aircraft was not known initially.

4 Summary and Conclusions

We have tried two different methods in filtering a SAR image. The first method filtered using a Haar wavelet. The wavelet did successfully smooth the image, but it did not help visually. In Figure 8 we saw that the wavelet added backscattering of the signal resulting in more bright streaks on an object. Due to the overwhelming 'unhelpfulness', it is not recommended to filter SAR images using wavelets for either purpose of human visualization or a network channel.

The second method attempted to filter the SAR images using the Robust PCA. This method was successful in darkening the background, but it also darkened the objects. If vehicles were close to the same color as the background, part of the vehicle representation was lost- as seen in Figure 10. In Figure 11 we saw that it did help in making the vehicles, helicopter, and

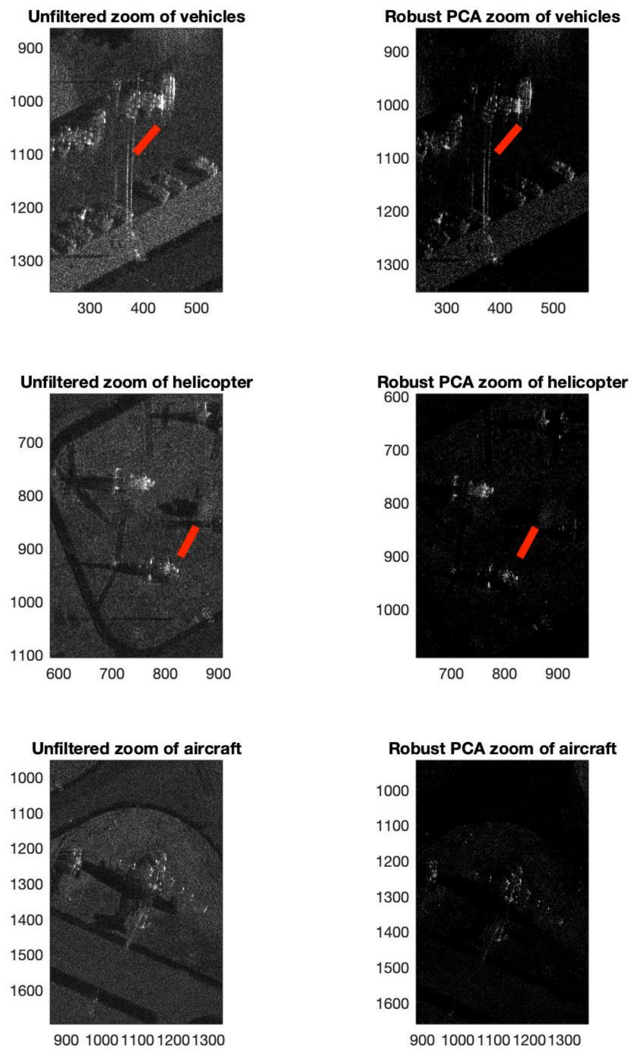

Figure 11: Comparing unfiltered and PCA images for vehicles, helicopters, and an aircraft.

aircraft 'pop' as the background was darkened, but losing the shadows didn't help in being able to identify the type of structure. This method could be used within a network channel, but it will most likely need more work before it provides results in image classification.

4.1 Path Forward

To understand where the Robust PCA would work, the correct weighting parameter for this type of problem should be found. Throughout these cases we chose $\lambda = 0.2$. However, as we have an image that is of size mxn, $\lambda = m^{-1/2}$ has been named[6]. When attempted, the images were not usable. A more suitable parameter between these two options might exist.

Although Singular Value Decomposition shows promise, using the Robust PCA method may be ill-fit for this type of problem. Perhaps a different SVD algorithm will provide better results.

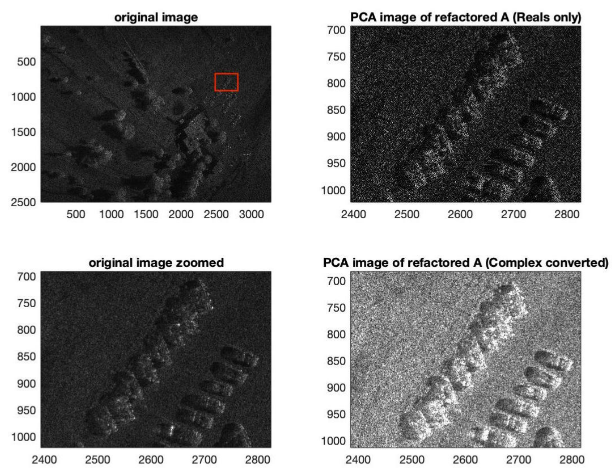

Figure 12: In 'original image' we have the original SAR image with a red box around where the other three images have zoomed in. The right column shows two different outputs from keeping the complex data during analysis.

The very last thing we will cover is our eleventh-hour refactoring as a proposal for a path forward with Robust PCA. Throughout this paper before running the `inexact_alm_rpca` algorithm, we subtracted the average. This attempt was to remove the background. Removing the average may be removing too much of the signal information we wish to retain. In an effort to kick start future work, we removed the subtraction of the average. We also went further and instead of bringing in the image with the complex data converted to reals, we kept the complex data and split within the function. Performing this extra step yielded positive results in possible paths for an additional channel into a deep learning network. Figure 12 shows this refactorization. When looking at this image, we plotted two different outputs that the refactored function produces. The top right is plotting the direct output from `inexact_alm_rpca`, which contains only real numbers. The bottom right is plotting both the real and imaginary parts. Either of these options may be more suited for placing within a channel, even if it does not completely remove the background.

References

[1] Charles V. Jakowatz, Jr. [et. al]. *Spotlight-mode synthetic aperture radar: a signal processing approach*, Springer, 1996.

[2] Sandia National Laboratories. "What is Synthetic Aperture Radar (SAR)?" `https://www.sandia.gov/radar/what_is_sar/`

[3] Walter G. Carrara [et. al]. *Spotlight synthetic aperture radar: signal processing algorithms*, Artech House, Inc., 1995.

[4] J. Nathan Kutz. *Data-Driven Modeling & Scientific Computation: Methods for Complex Systems & Big Data*, Oxford University Press, 2013.

[5] Weisstein, Eric W. "Haar Function." From MathWorld–A Wolfram Web Resource. `http://mathworld.wolfram.com/HaarFunction.html`

[6] Zhouchen Lin, Risheng Liu, and Zhixun Su, *Linearized Alternating Direction Method with Adaptive Penalty for Low Rank Representation*, NIPS 2011. `https://arxiv.org/abs/1009.5055`

Appendices

Please refer to my Github repo for all code and related documents.

A Matlab Functions

Code was provided from Sandia National Laboratories for loading SAR images. Not placed here, but is linked in Github.

```
function [Image_AR, svdIm_u, svdIm_s, svdIm_v] =
 svdImage(Image,lambda)
% This function brings in an image and a weighting parameter and
% runs inexact_alm_rpca on it.
%   Inputs:
%       Image matrix (must be a matrix of real numbers)
%       lambda (must be a positive real number)
%   Outputs:
%       Image_AR: A (Real) matrix solved from inexact_alm_rpca.
%       svdIm_u, svdIm_s, svdIm_v: SVD components of matrix A where
%       A = u * s * v'

[~,n] = size(Image);
mn = mean(Image,2)/2;
Image_resize = Image - repmat(mn,1,n);
[Image_AR,~] = inexact_alm_rpca(Image_resize.',lambda);
[svdIm_u, svdIm_s, svdIm_v] = svd(Image_AR');


end
```

Published with MATLAB® R2019b

```
function [Image_AR, R1, svdIm_u, svdIm_s, svdIm_v] =
 svdImage_refactored(Image,lambda)
% This function brings in an image and a weighting parameter and
% runs inexact_alm_rpca on it.
%   Inputs:
%       Image matrix (complex numbers),
%       lambda (must be a positive real number)
%   Outputs:
%       Image_AR: A Real matrix solved from inexact_alm_rpca (i.e., A
 = A+0i)
%       R1: Real and Imaginary parts of matrix A (i.e., A = A+Ai)
%       svdIm_u, svdIm_s, svdIm_v: SVD components of Real matrix A
 where
%       A = u * s * v'

ur = real(Image);
ui = imag(Image);

[Image_AR,~] = inexact_alm_rpca(real(ur.'),lambda);
[Image_ARi,~] = inexact_alm_rpca(real(ui.'),lambda);

R1 = Image_AR + i*Image_ARi;

[svdIm_u, svdIm_s, svdIm_v] = svd(R1.');

end
```

Published with MATLAB® R2019b