EDGE-INFORMED FEATURE DETECTION

NATHAN SMITH

Abstract. A new algorithm is developed to extract a feature registration that effectively incorporates information from Fourier data, succinctly points to relevant areas within an image, and exhibits robustness to noise and missing data of various types. We focus four challenges inherent to many Fourier imaging applications that hinder accurate feature extraction from typical inverse Fourier image recovery. Our new method is designed to isolate target areas in an image by using a concentration method factor to identify edges in the frequency domain. Almost all feature extractions tested yielded improved precision. Data loss is largely overcome as the Fourier edge detection method behaves excellently when standard spatial domain edge detectors fail. Numerical examples are provided to demonstrate the accuracy and robustness of this novel idea.

1. Introduction. Accurate feature recognition is important in many applications including medical imaging and remote sensing with Synthetic Aperture Radar (SAR). While there is no formal definition of an image feature, existing algorithms extract various key details from images including traces of lines, regions, and segments. In practice, a feature can be represented as any set of information that provides insight into the image's ground truth or distinguishes what in the image is important from what may be extraneous.

While many "off the shelf" feature extraction algorithms are effective given sufficient input data, several issues render direct application impracticable, including vulnerability to missing or corrupted data or lack of robustness to instrumental noise. Most importantly, no methods involve direct analysis of images when the measurement instruments acquire noisy and possibly corrupted Fourier data. Problematically, both MRI and SAR phase history data can be described in this way. Then it is inevitable that the information living in the frequency domain often disappears or distorts in a spatial representation.

We are interested in developing a method to extract a feature registration that **(1)** effectively incorporates information from Fourier data **(2)** succinctly points to relevant areas in an image and **(3)** exhibits robustness to noise and missing data. This investigation proposes a new technique for extracting features from images represented by Fourier data. To ensure accuracy, efficiency, and robustness, our new method uses the Fourier concentration factor edge detection method, first developed in [10]. Almost all feature extractions tested yielded improved precision, and data loss was largely overcome as the Fourier edge detection method behaves excellently when standard spatial domain edge algorithms fail.

The rest of this thesis is organized as follows. In Section 2 we include some important background information needed to develop our new algorithm. In Section 3 we discuss how we plan to detect edges and use them in the downstream process of feature extraction. In Section 4 we tune our novel method and test it against multitudinous corruption modes to demonstrate its efficacy and robustness. Finally, concluding thoughts and ideas for future work are provided in Section 5.

2. Background Information. Feature extraction is a fundamental operation in image processing, enabling the comparison, alignment, and interpretation of visual data across applications ranging from medical imaging to remote sensing. Accurate feature registration becomes especially critical in scenarios where images differ due to missing information, illumination variations, or inherent noise. Synthetic Aperture Radar (SAR) imaging presents a particularly challenging case for feature extraction, as SAR images differ from optical images in both their noise characteristics and the structure of their underlying data.

Traditionally, feature extraction algorithms operate directly in the spatial domain (Euclidean), detecting distinctive patterns such as corners, edges, or blobs within the intensity map of an image. Techniques such as the Scale Invariant Feature Transform (SIFT), Speeded-Up Robust Features (SURF), and Oriented FAST Rotated BRIEF (ORB) have been extensively applied in this setting, offering robust methods for identifying and encoding salient image features. However, these methods are limited by their reliance on local pixel-based information, which may be degraded or obscured in images characterized by non-Gaussian noise or imaging artifacts, as is common in SAR data. Moreover when the data are acquired as Fourier samples, as in the case for SAR and MRI, the data must be processed first to obtain an image. The situation is exacerbated in conditions where the data might be corrupted or otherwise unreliable in certain frequency bands.

The goal of this research is to investigate whether integrating edge information derived from its collected Fourier samples can enhance feature extraction, particularly in contexts where traditional spatial domain algorithms struggle. In particular, this project examines methods for extracting edge locations using Fourier concentration factors, with the broader aim of improving the alignment and matching of images in applications such as SAR imaging.

2.1. Standard Feature Extraction Algorithms from Digitized Images.

There are too many "standard" image feature extraction algorithms to reasonably explain and demonstrate thoroughly in this thesis. However, as our goal is to identify key targets given a certain image, it is instructive to evaluate the performance of prominent existing feature extraction algorithms on a meaningful test problem. Because we are motivated by SAR imaging applications, we use a SAR image for this purpose (see left image in Figure 4). To generate a realistic data environment, zero mean Gaussian noise with different SNR values is added the original SAR image, along with the features we seek to identify (building, cars, and ship) and distinguish from the background.

Three main existing algorithms for computing keypoints in images are the Scale Invariant Feature Transform (SIFT), the Speeded Up Robust Features (SURF), and the Oriented FAST and Rotated BRIEF (ORB) methods. I will provide a brief overview of how these algorithms work.

SIFT: A widely used classical feature detection method is the **Scale-Invariant Feature Transform** proposed by David Lowe in 2004 [8][14]. SIFT detects distinctive local keypoints in an image and computes feature descriptors that are robust to common image transformations. In particular, SIFT features are invariant to image scale and rotation and partially invariant to illumination and viewpoint shifts [14]. This makes SIFT useful for matching features between images when they are at different zoom levels, rotations, or lighting schemes [14]. The algorithm achieves these properties through a series of steps.

SIFT identifies candidate keypoints by searching for local extrema in a multi-scale representation of the image which is progressively blurred and subtracted using a Difference-of-Gaussian approach

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * f(x, y) = L(x, y, k\sigma) - L(x, y, \sigma), \tag{2.1}$$

to create a scale-space in which interest points can be identified across different scales [14]. Here σ represents a blurring level scaled $k \in 1, \dots, K$ times. Keypoints found this way associate with a characteristic scale giving them scale invariance.

Each candidate is then refined: SIFT filters out unstable point such as those with low contrast or those lying along edges, since these are less reliable [8]. By discarding keypoints that are too edge-like or noise sensitive, the algorithm guarantees repeatability of remaining keypoints.

For surviving keypoints, SIFT assigns a consistent orientation based on the local image gradients. In a neighborhood around the keypoint, the algorithm computes a gradient (with gradient magnitude $m(x, y)$) orientation (with orientation $\theta(x, y)$) histogram and chooses the dominant orientation(s) [8], computed as

$$\begin{aligned} m(x,y) &= \sqrt{(L(x+1,y)-L(x-1,y))^2+(L(x,y+1)-L(x,y-1))^2} \\ \theta(x,y) &= \tan^{-1}((L(x,y+1)-L(x,y-1))/(L(x+1,y)-L(x-1,y))) \end{aligned} \tag{2.2}$$

Observe that the image patch is conceptually rotated to align with the dominant direction, providing rotational invariance. Specifically, the descriptor computed for the keypoint will be oriented relative to the keypoint's predominant local orientation so that rotation of the image yields the same descriptor values [14].

SIFT generates a descriptor that encodes the local image structure around each keypoint. After scaling and rotating the neighborhood to canonical orientation, the algorithm measures the distribution of local gradients (edge orientations) within the patch [8]. The gradient orientations are summarized in a set of historgrams [14]. The discriptor is normalized to withstand illumination changes. The resulting representation is a rich keypoint encoding that contains local texture pattern that can enable reliable matching between multiple images. SIFT's effectiveness and robustness have made it a cornerstone of computer vision with adaptations made for SAR [8].

SURF: In 2006, the Speeded-Up Robust Features algorithm was introduced [11] as a local feature detector and descriptor inspired by SIFT but optimized for greater speed. SURF employs a blob detector based on the Hessian matrix using the determinant of the Hessian as the measure of interest point strength (this is called the "Fast-Hessian" detector). To achieve computational efficiency, SURF approximates Gaussian second order derivatives with square-shaped box filters and uses integral images to rapidly evaluate these filters at any scale.

For example, the SURF detector uses a 9×9 box filter to approximate a Gaussian with $\sigma \approx 1.2$ at the initial scale; the filtered outputs (denoted D_{xx}, D_{yy}, D_{xy} for second-order derivatives in the x, y and xy directions) are combined to compute the approximate Hessian determinant as $\det(H) \approx D_{xx}, D_{yy} - (D_{xy})^2$ The SURF descriptor component then summarizes the local intensity distribution using Haar wavelet responses in the neighborhood of each keypoint, which are also efficiently calculated via the integral image technique. SURF yields repeatable and distinctive features with robustness comparable to SIFT while running much faster [13].

ORB Oriented FAST and Rotated BRIEF is a local feature detector introduced by Rublee *et al.* (2011) as a fast and efficient alternative to classical algorithms like SIFT and SURF [6]. ORB combines the FAST corner detector with an oriented variant of the BRIEF descriptor; it detects keypoints using FAST and then asigns each keypoint a consistent orientation based on the intensity centroid of its local image patch using image moments [20]. The BRIEF descriptor is subsequently rotated according to this keypoint orientation, producing a binary feature vector that is invariant to in-plane rotation [20]. By design, ORB achieves comparable matching performance to SIFT with fractional computational cost and boasts robustness to noise [6]. It has been wiedely adopted as an efficient alternative in real-time vision applications, and has been even adapted for use in SAR imagery. An ORB-based method for SAR image matching demonstrated strong robustness to speckle noise and outperformed SIFT in such scenarios [7].

Figure 1 illustrates how the three feature algorithms mentioned in this section identify keypoints on our test problem. Their performance is demonstrated on the original image, uncorrupted, as well as on a version of the image whose data as given in the frequency domain is obstructed along radial lines such as in .

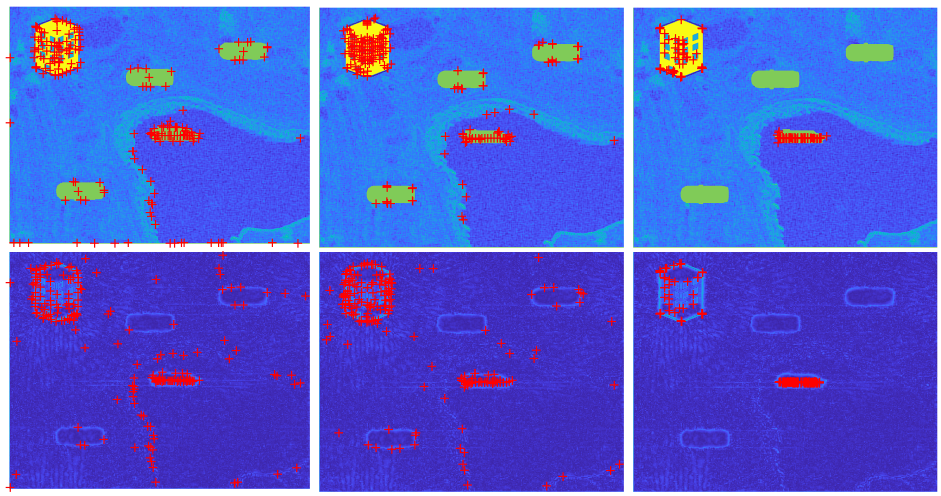

Fig. 1: Top 200 keypoints detected by SIFT, SURF, and ORB before (top row) and after (bottom row) radial Fourier corruption.

As SIFT is the top performer in our cursory image tests and widely considered to be the gold standard keypoint algorithm, we will choose SIFT as the baseline moving forward in the thesis.

2.2. Standard Edge Detection Algorithms. We are interested in the hypothesis that recovering the digitized image from the given Fourier data is not necessary for feature extraction, but rather that the algorithms mentioned above only require the corresponding *edge map*, since it appears that the keypoints detected by front-running feature-finders congregate around the image's edges. Therefore, we may be able to recover equally useful information about the underlying ground truth without requiring the entire image. This is important because recovering an image from noisy and corrupted Fourier data is less accurate, less efficient, and less robust than recovering its edges [23].

The immediate option is then to apply the aforementioned feature recoveries to edge maps generated by applying standard edge detections algorithms on images digitized out of the frequency domain. Several such edge detectors exist, and below I will introduce the a few.

In general, in the spatial domain, edge detection is performed pixel-wise using kernels that amplify jump discontinuities in magnitude. The simplest kernel for edge detection is called Roberts Cross, which applies 2×2 differencing kernels in two dimensions. The kernels appear as follows:

$$G_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \qquad G_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

The gradient magnitude is given by

$$|G| = \sqrt{G_x^2 + G_y^2}$$

and approximated as

$$|G| \approx |G_x| + |G_y|$$

for quicker computation. Given pixel grid:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

The corresponding magnitude of the edge-mapped output is given by $|G| = |a - d| + |b - c|$ and slotted into the the top left at pixel a.

Roberts Cross edge detection is the simplest and the easiest to compute [21]. However, its small kernel size leaves it vulnerable to image noise, it requires a large separation of scales to generate edges robust to thresholding, and it lacks optional parameters for tailoring edge recognition to the specific case [17]. subsection 2.2 (top right) demonstrates some of these pitfalls. Here we begin with a 256×256 pixelated noisy image,

$$Im_{noise} = Im + \boldsymbol{\eta}, \tag{2.3}$$

where $\boldsymbol{\eta} \sim \mathcal{N}(0, \Sigma)$, and covariance matrix Σ is determined by the SNR. In subsection 2.2, the noisy image (top-left) is generated with $\Sigma = \sigma^2 I$ and $\sigma = .1$.

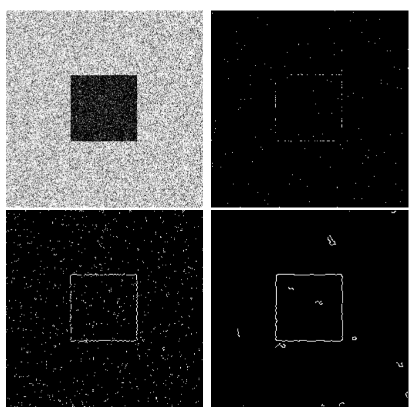

Fig. 2: (top-left) Given noisy data (2.3); (top-right) Roberts Cross edge map; (bottom-left) Sobel edge map; and (bottom-right) Canny edge map.

Similar to the Roberts Cross kernel is the Sobel kernel [18]:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}, \qquad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

Computation for the Sobel edges follows the same procedure as for the Roberts Cross kernel, but the larger 3×3 kernel comes with a few advantages. With the larger kernel whose output slots in the center, the computation better blurs the effect of noise than the Roberts Cross method [18]. The Sobel operator approximates the spatial gradient of image intensity by convolving the image with two 3×3 kernels, G_x and G_y, corresponding to the horizontal and vertical directions . The gradient magnitude at each pixel is then given by

$$G(x, y) = \sqrt{(G_x * Im)^2 + (G_y * Im)^2},$$

which quantifies the local rate of intensity change and identifies edges. Since the output magnitude relates to a gradient, the resulting edge map better reports the sharpness of each discontinuity. This also ensures that any point of constant magnitude in the input image translates to a zero value in the edge map [18][21].

Finally, Canny edge detection is another popular edge finding algorithm. First, the algorithm reduces image noise using a 5×5 Guassian filter. Then, it applies Sobel, as seen before, where:

$$\text{Edge\_Gradient}(G) = \sqrt{G_x^2 + G_y^2}, \qquad \text{Angle}(\theta) = \tan^{-1}\left(\frac{G_x}{G_y}\right). \tag{2.4}$$

Each pixel is then checked against the neighboring pixels on both sides along the gradient line to determine whether the center pixel is a local maximum, and the pixel is suppressed if it is not a local maximum. Finally, through a process called Hysteresis thresholding, pixels the algorithm is uncertain are edges are determined to belong to edges if they lie near edge pixels the algorithm is more certain about. [19]

Figure 3 illustrates how the SIFT algorithm is applied to the edge maps produced by the Roberts Cross, Sobel, and Canny edge maps on data obtained AS HOW. The top 200 SIFT keypoints are the red crosses superimposed onto each corresponding edge map. These edges are recovered from pixelated image data reconstructed from an uncorrupted Fourier sample.

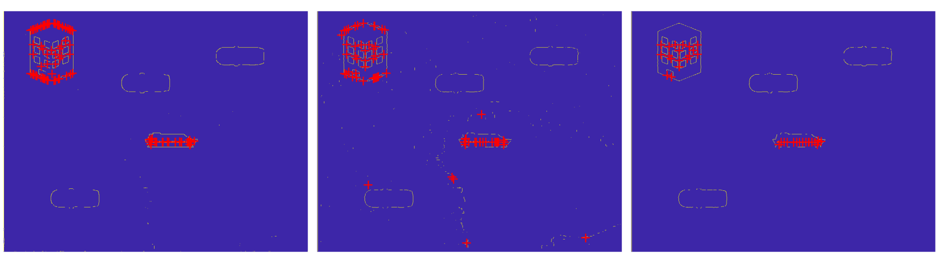

Fig. 3: On each image, the top 200 SIFT keypoints are overlaid as red crosses. The **leftmost** image is a Roberts Cross edge map of our test image. The **middle** is Sobel and the **right** is Canny. The canny hysteresis thresholding parameter is set as $[0.2, 0.4]$.

As seen in Figure 3, visually apparent failures plague these edge maps. Since Roberts Cross and Sobel have similar constructions, it makes sense that their edge maps are similar; unfortunately, both capture thin edges, unfit for our desires, and gaps exist where edges should represent closed objects. In addition, SIFT, which is our strongest feature identifying candidate so far, fails to identify three prominent shapes.

2.3. Images in the Fourier Domain. In some applications, such as Synthetic Aperture Radar (SAR) and Magnetic Resonance Imaging (MRI), data are acquired as Fourier samples. Assume that the underlying image $f(x, y)$ in on the domain $[-1, 1] \times [-1, 1]$.[1] The corresponding N Fourier samples are then

$$\hat{f}(k,l) = \frac{1}{4}\int_{-1}^{1}\int_{-1}^{1} f(x,y)e^{-i\pi(kx+ly)}dxdy, \qquad -\frac{N}{2} \leq k,l \leq \frac{N}{2}-1. \tag{2.5}$$

More precisely, the given data is of the form

$$\hat{b}(k,l) = \hat{f}(k,l) + \hat{\epsilon}(k,l), \qquad -\frac{N}{2} \leq k,l \leq \frac{N}{2}-1, \tag{2.6}$$

where $\hat{\boldsymbol{\epsilon}} \sim \mathcal{CN}(0, \Sigma)$ is complex circularly symmetric zero-mean Gaussian noise with covariance matrix Σ, typically taken to be diagonal with constant entries when the noise is assumed to be uncorrelated and homoscedastic across frequencies. We also note that instrumentation failure may lead to $\hat{b}(k, l) = \hat{\epsilon}(k, l)$ (meaning that no underlying signal information was collected or is otherwise unusable) in some frequency bands. Since lower frequencies encode smooth global structures and high frequency components capture finer details like edges and sharp gradients, careful consideration must be given to how (2.6) can be exploited to isolate meaningful features, suppress noise, and analyze image geometry.

Note that the standard feature extraction algorithms described earlier would require a preprocessing step to recover (2.3), obtained by

$$f_N(x,y) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}\sum_{l=-\frac{N}{2}}^{\frac{N}{2}-1} \hat{b}(k,l)e^{i(kx+ly)}, \tag{2.7}$$

at a specific set of grid points $x_j, y_m \in [-1, 1] \times [-1, 1]$. Even with filtering this preprocessing step of image recovery can cause issues such as image artifacts, loss of sharp edges (if the coefficients are heavily filtered), or misrepresentations of feature shapes or intensities, issues which ultimately can undermine downstream feature extraction algorithm. These issues are exacerbated if the collected samples are corrupt or missing.

Discretized model of Fourier coefficients. Since this investigation uses a SAR image as its test case (see Figure 4), we provide the corresponding discrete Fourier transform (DFT) matrix [12]

$$F(k,l) = \sum_{j=0}^{N-1}\sum_{m=0}^{N-1} f(x_j,y_m)e^{-\imath 2\pi(\frac{kj}{N}+\frac{lm}{N})}, \qquad -\frac{N}{2} \leq k,l \leq \frac{N}{2}-1, \tag{2.8}$$

where $x_j = -1 + \frac{2j}{N}$ and $y_m = -1 + \frac{2m}{N}$, $j, m = 0, \ldots, N-1$. Our numerical experiments use (2.8) on the underlying noisy image Im_{noise} in (2.3) with different SNR levels. Specifically we define

$$B(k,l) = \sum_{j=0}^{N-1}\sum_{m=0}^{N-1} Im_{noise}(j,m)e^{-\imath 2\pi(\frac{kj}{N}+\frac{lm}{N})}, \qquad -\frac{N}{2} \leq k,l \leq \frac{N}{2}-1. \tag{2.9}$$

[1] The domain boundaries are only defined for convenience, as a linear transformation allows for any $x \in [a, b]$, $y \in [c, d]$.

2.4. Potential Issues in Frequency Domain Images. In practice, we may not have data for every frequency in our spectrum. For example, radar instruments operating concurrently alongside other systems using similar frequencies may experience data corruption or may be prohibited from transmitting frequencies regulated by frequency management agencies [16]. Moreover, we are interested in long-term compression of data; therefore, it is instructive to know how little data we can store and still obtain useful information about the underlying truth behind an image.

You can observe in Figure 4 the visual effects of two specific kinds of corruption that can occur when collecting Fourier samples. The digital reconstructions are shown beside an image digitized from an uncorrupted sample.

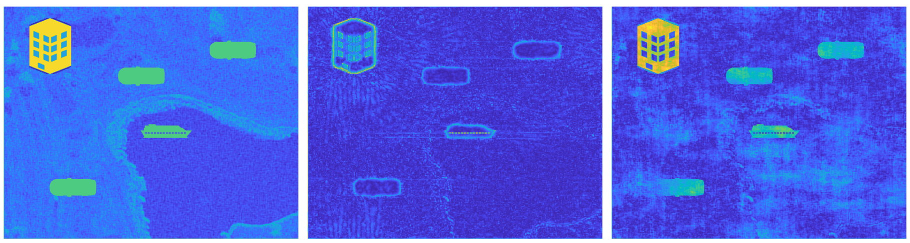

Fig. 4: The **leftmost** image displays the uncorrupted base test image [5][23]. In the **middle**, 12 radial lines are missing, and on the **right** a random sample of the Fourier coefficients disappeared. This can have deleterious effects on Standard edge detection.

In this investigation we consider the removal of some radial lines in the given Fourier data (2.9). This can be accomplished by first choosing J angles $\theta_j = \frac{j}{J}\pi \;\; j \in 0, \ldots, J$. We then apply `fftshift` to the image so the zero frequency component moves to the center of the Fourier image. From here we define frequency component matrices

$$\mathbf{K} = \begin{bmatrix} -\frac{N}{2} & \cdots & \frac{N}{2}-1 \\ \vdots & \vdots & \vdots \\ -\frac{N}{2} & \cdots & \frac{N}{2}-1 \end{bmatrix} \quad \text{and} \quad \mathbf{L} = \begin{bmatrix} -\frac{N}{2} & \cdots & -\frac{N}{2} \\ \vdots & \vdots & \vdots \\ \frac{N}{2}-1 & \cdots & \frac{N}{2}-1 \end{bmatrix}, \tag{2.10}$$

leading to J *radial mask matrices*

$$R(\theta_j) = \mathtt{abs}(\cos(\theta_j)\mathbf{K} + \sin(\theta_j)\mathbf{L}), \quad j \in 1, \ldots, J. \tag{2.11}$$

Finally, to simulate missing frequency bands in applications such as SAR, we mask radial lines in our frequency domain image data. If f is an image in the frequency domain, then

$$B_j^{mask}(k,l) = \begin{cases} F(k,l) & \text{if} \quad R(\theta_j)) \geq 1 \\ 0 & \text{if} \quad R(\theta_j) < 1, \end{cases} \tag{2.12}$$

Where B_j^{mask}, $j = 1, \ldots, J$ gives the radially masked image missing J radial lines of frequency data. Then our radially masked Fourier data is generated iteratively as

We apply the mask B_J^{mask} iteratively over $F_0(k,l) = F(k,l)$ where if

$$F_j(k,l) = F_{j-1} \odot B_j^{mask} \text{ for } j = 1, \ldots, J$$

Then $F_{rad} = F_J$ where F_{rad} denotes the final radially corrupted image.

Our remote sensing instrument may instead be damaged such that it relinquishes data randomly, and we can simulate this disastrous scenario easily by initializing a random matrix defined below. If Im has size $N \times N$, the matrix is given by

$$\mathbf{R} \sim \mathcal{U}(0,1)^{N \times N} \tag{2.13}$$

Where $u \sim \mathcal{U}(0,1)$. We choose some number $\rho \in (0,1)$ to say

$$M(k,j) = \begin{cases} 1 & \text{if} \quad R(k,l) > \rho \\ 0 & \text{otherwise} \end{cases} \tag{2.14}$$

We then apply this mask to simulate frequency-domain dropout:

$$F_M(k,l) = M_{k,l} \odot F(k,l), \tag{2.15}$$

where $\odot$ means componentwise multiplication. This models a scenario where a proportion ρ of frequency coefficients are lost. The resulting corrupted spectrum $\hat{f}_{\text{corrupted}}$ can then be used to investigate how edge formation degrades under random measurement loss.

We will talk in the next section about how edges manifest in frequency space, a development fundamental to many modern edge detection and reconstruction methods.

2.5. Concentration Factor Based Edge Detection. The methods discussed in this section are based on [23] and [9]. Edge detection for images can be conducted by looking at the jump discontinuities of piecewise smooth functions $f(x,y)$. We can reasonably assume that edges occur along a finite collection of smooth (often closed) curves in the spatial image domain.

We first study a one-dimensional case. Let $f : [0,1] \to \mathbb{R}$ be a piecewise smooth function with $\mathcal{M}$ jump discontinuities at locations $\{\xi_m\}_{m=1}^{\mathcal{M}}$. Our N Fourier coefficients are given by

$$\hat{f}_k = \int_0^1 f(x) e^{-\imath 2\pi k x} dx, \quad k = -\frac{N}{2}, \cdots, \frac{N}{2} - 1 \tag{2.16}$$

And the jump function is

$$[f](x) = f(x^+) - f(x^-), \tag{2.17}$$

with $f(x_0^+) = \lim_{x \to x_0}^{+} f(x)$ and $f(x_0^-) = \lim_{x \to x_0}^{-} f(x)$. The jump function can then be written

$$[f](x) = \sum_{m=1}^{n} [f](\xi_m) I_{\xi_m}(x) \tag{2.18}$$

where

$$I_{\xi_m}(x) = \begin{cases} 1 & \text{if} \quad x = 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.19}$$

From our Fourier samples given in (2.16), [23] defines the concentration factor edge detection method as

$$S_N^{\sigma} f(x) = \imath \sum_{1 \le |k| \le \frac{N}{2}} \operatorname{sign}(k) \sigma\left(\frac{|k|}{N}\right) \hat{f}_k e^{\imath 2\pi k x} \tag{2.20}$$

Where, for our purposes, $\sigma\left(\frac{|k|}{N}\right) = 2\pi\imath \cdot |k| \cdot \sqrt{\frac{\pi}{\sigma}} \cdot \exp\left(-\frac{\pi^2}{\sigma}|k|^2\right)$ denotes the exponential concentration factor as defined in [1]. This is smooth and rapidly decaying to suppress high frequency noise.

2.6. Edge Detection for Images from Fourier Samples. Since images are two dimensional and discrete, there are infinite jump discontinuities, and the discontinuities are not pixel-wise unique but are instead dependent on the direction at which you measure the jump. Therefore we instead look for jumps in the normal direction of the edge curve.

Definition [23] Assume the discontinuities of f form finite closed, smooth edge curves $\Gamma_m, 1 \le m \le \mathcal{M}$. For a point $(x, y) \in \Gamma_m$ at a discontinuity, let $\mathbf{n}(x, y)$ be normal to (x, y) with respect to the edge curve Γ_m. Consider the corresponding one dimensional continuous function $g(t) = f(x, y) + t\mathbf{n}(x, y)$. The jump function is then defined as

$$[f](x, y) = [g](0), \quad (x, y) \in \Gamma_m, 1 \le m \le \mathcal{M} \tag{2.21}$$

In this thesis, we assume we are given uniformly spaced Fourier samples (2.5). We want $[f](x, y) I_\Gamma(x, y)$ such that I_Γ is the indicator similar to (2.19) defined for one edge curve. We parameterize the Γ as

$$x = u(s), \ \ y = v(s), \quad s \in [a, b] \tag{2.22}$$

where $u(s), v(s)$ are smooth and $[a, b]$ is some region of the image.[2] Let $\theta(s)$ be the normal direction to $(u(s), v(s))$ such that

$$x = u(s) + r\cos\theta(s), \ \ y = v(s) + r\sin\theta(s) \tag{2.23}$$

parameterizes the points around Γ and r is a small range $r \in [-\epsilon, \epsilon]$ with $\epsilon > 0 \in \mathbb{R}$ is small.

The indicator function I_γ has support of measure zero, so we must apply a regularization $h(\frac{r}{\epsilon})$ defined on the small closed interval $[-\epsilon, \epsilon]$ and satisfying $h(0) = 1$. As in [23], we use

$$h\left(\frac{r}{\epsilon}\right) = \exp\left(-5\left(\frac{r}{\epsilon}\right)^2\right) \tag{2.24}$$

Giving regularized edge function

$$h(x, y) = [f](u(s), v(s)) h\left(\frac{r}{\epsilon}\right) \tag{2.25}$$

Given data as a Fourier partial sum, our Fourier partial sum approximation of $h(x, y)$ is written similarly to (2.8)

$$h(x, y) \approx h_N(x, y) = \sum_{k=-N/2}^{N/2-1} \sum_{l=-N/2}^{N/2-1} H(k, l) e^{2\pi\imath(kx + ly)}, \tag{2.26}$$

[2] We use $[a, b]$ just to define the relationship between $H(k, l)$ in (2.27) and (2.28) but the values themselves are arbitrary and not used in computation.

with coefficients

$$H(k,l) = \int_a^b [f](u(s),v(s))(v'\cos\theta - u'\sin\theta)\epsilon\hat{h}(\epsilon(k\cos\theta + l\sin\theta))ds, \tag{2.27}$$

where $\hat{h}(\cdot)$ are the Fourier coefficients of (2.24). Using the parameterization given by (2.23), we can approximate $\hat{f}(k,l)$ as

$$\hat{f}(k,l) \approx \int_a^b \frac{[f](u(s),v(s))(v'\cos\theta - u'\sin\theta)}{2\pi\imath(k\cos\theta + l\sin\theta)} e^{-2\pi\imath(ku+lv)}ds \tag{2.28}$$

As done in [23] we fix $\theta = \theta_r$ so that

$$\tilde{H}_{\theta_r}(k,l) = 2\pi\imath(k\cos\theta_r + l\sin\theta_r)\epsilon\hat{h}(\epsilon(k\cos\theta_r + l\sin\theta_r))\hat{f}(k,l).$$

This allows us to construct R edge masks for our image where our R rotation angles are given by

$$\theta_r = \left(\frac{r-1}{R-1}\right)\pi, \quad r = 1,\dots,R, \tag{2.29}$$

Finally, we compute R partial sum approximations at $\theta_r,\ r = 1,\dots,R$, using (2.28) inside the sum by (2.8). Our final edge map is created by averaging over R to find $E_R \in \mathbb{R}^{N\times N}$

$$E_R(x_j,y_m) = \frac{1}{R}\sum_{r=1}^{R} H_{\theta_r}(x_j,y_m), \quad j,m = 1,\dots,N. \tag{2.30}$$

This approach offers a mathematically grounded and rotation-aware alternative to classical gradient-based edge detectors, particularly useful when operating directly on Fourier data or under conditions where the spatial domain is inaccessible or corrupted.

3. Main Idea. This thesis introduces a novel pipeline for feature extraction directly from Fourier-domain image data. The goal is to derive a representation of stable, spatially localized features that (1) incorporates information available in the frequency domain, (2) suppresses unstable or noisy edges, and (3) remains robust under realistic corruption modes commonly found in SAR and other indirect imaging techniques. Inspired by [23], we begin by constructing a ground truth image Im composed of natural background scenes and a small set of embedded objects (e.g., vehicles, ships, buildings). Each image is formed by overlaying scaled and shifted versions of object templates onto a fixed background image.

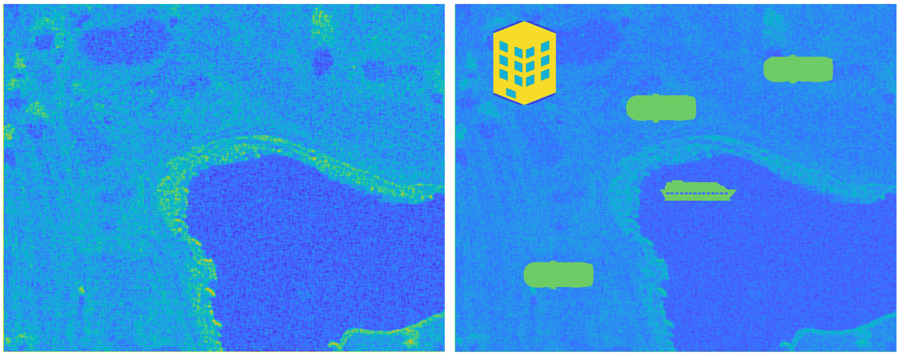

Fig. 5: **left** An aerially captured image of a golf course; **right** Five objects, including three tanks (magnitude = 1), a ship (magnitude = 1), and a building (magnitude = 1.5) are added to the underrlying image.

Figure 5 displays SAR images borne and shaped in the spatial domain, stored and represented pixel-wise as a grid or matrix of magnitudes. We are not exactly interested in this data, however, as tackling our goals from images represented by data in the frequency domain presents a much richer challenge and a problem far more immediately reflective of the challenges inherent to SAR and MRI applications. Therefore, the setup of our toy problem continues.

First, the image Im is shifted and transformed to the frequency domain as

$$\hat{f} = \mathcal{F}(Im_{\text{true}}) = \texttt{fft2(ifftshift(}Im_{\text{true}}\texttt{))}. \tag{3.1}$$

We assume the data were provided in this way, and from now on we make no use of the original spatially described canvas. We have options for several corruption models to the Fourier data, yielding multiple problems in the same image.

1. **We do nothing:** The data, having undergone the Fourier transform in (3.1) now provides a baseline off which to evaluate the robustness of various techniques to the following forms of corruption.
2. **Additive Complex Gaussian Noise:** Simulating thermal or instrumental noise with controlled SNR.
3. **Random Missing Coefficients:** Emulating incoherent or partial acquisition patterns.
4. **Radial Line Removal:** Mimicking limited-angle data in SAR or missing look directions.
5. **Low Magnitude Features:** Simulating disguised targets or non reflective surfaces.

In order to visually check our progress, we feed the corrupted frequency data into the inverse-transformed shown below:

$$Im_{\text{corrupted}} = \left| \mathcal{F}^{-1}\{\hat{f}\} \right|. \tag{3.2}$$

These corruptions, commonplace in aforementioned applications, can amount to catastrophic information loss as demonstrated earlier in Figure 4. We explore the idea that the corrupted data left in the frequency domain contains better information than its digitized counterpart.

3.1. Frequency-Based Edge Detection. Since edge information plays an integral role in accurate feature representations but that locations of edges can be lost in digitization, we instead extract edges from the corrupted Fourier data using the spectral differentiation method based on the concentration factor framework from Subsection 2.6. Directional derivatives of the regularized edge function are computed in the frequency domain and integrated (2.28). The edge responses are averaged over a discrete set of R directions $\theta_r \in [0, \pi]$ $r = 1, \dots, R$ given by (2.29) to estimate jump magnitude. Let $\hat{f}(k,l)$ denote the possibly corrupted 2D Fourier transform of the image $f(x,y)$ where $k, l \in \mathbb{Z}$.

For each θ_r an edge estimation is computed as a directional derivative along θ and transformed back with (3.2)

$$E_\theta(x,y) = \mathcal{F}^{-1}\left[\sigma(\xi_\theta) \cdot \hat{f}(k,l)\right](x,y) \tag{3.3}$$

Finally, we average over a finite set of R angles $\{\theta_r\}_{r=1}^{R} \in [0,\pi]$:

$$\mathcal{E} = \mathrm{Edge}(x,y) = \frac{1}{R}\sum_{r=1}^{R} |E_{\theta_r}(x,y)| \tag{3.4}$$

The result is therefore a map $\mathcal{E}$ capturing edge strength in the spatial domain. (Explained extensively in Subsection 2.6)

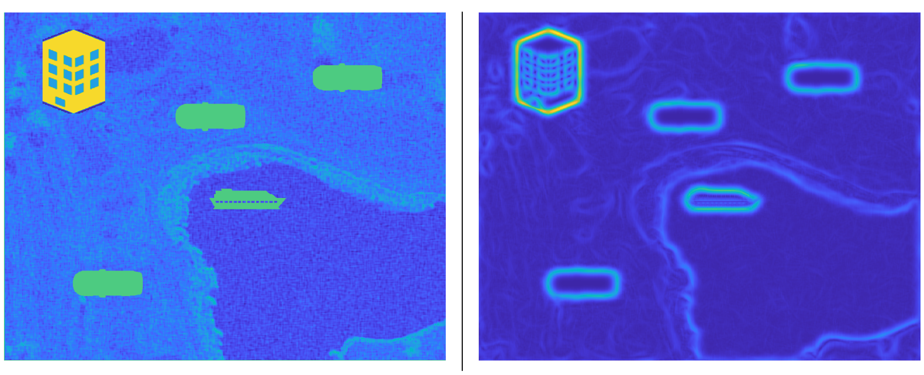

Fig. 6: On the **left** see the corrupted frequency domain image visualized here in its form digitized from (3.2). On the **right** see its edges computed directly from their Fourier representations and similarly digitized.

Since a spatially informed edge detector is relatively meaningless in the frequency domain, if we choose to proceed with its output we would be treating frequency components as if they were pixel intensities. Fourier coefficients encode global frequency content, so exploiting directional spectral differentiation and concentration factors enables us to find stronger edges, evident in Figure 6. As a bonus, we can repeat this process over multiple rotations to enhance the robustness of detection.

3.2. Region Extraction and Filtering. Because the resulting edge maps represent closed shapes when represented spatially, we can isolate the key features of the images. That is, morphological closure and hole filling are applied to form connected regions. We then extract geometric properties using connected component analysis. Very small blobs (e.g., noise or artifacts) and excessively large ones (e.g., background bleed or artifacts of under-smoothing) are discarded. The remaining regions represent spatially coherent and statistically stable features in the noise realizations.

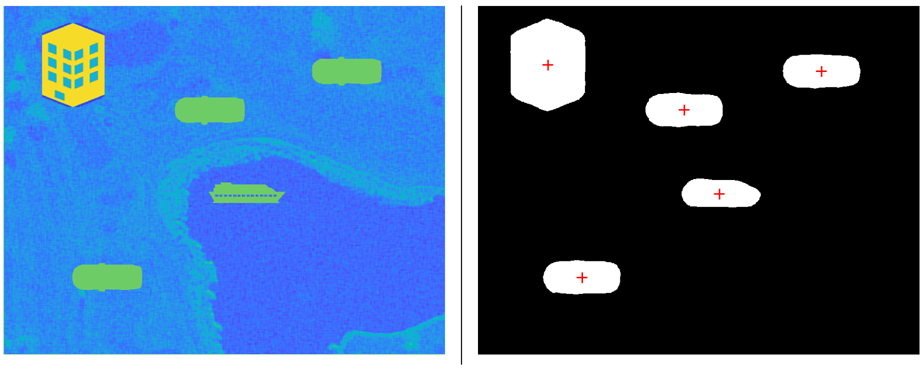

Fig. 7: On the **left** is the original base image also shown in Figure 5, and on the **right** appears the regions identified and filled inside the edges

To achieve the result in Figure 7, the edge map $\mathcal{E}(x, y)$ is thresholded—note that we are now working in the spatial domain—by a threshold $\tau_{\mathcal{E}}$ defined as

$$\tau_{\mathcal{E}} = \texttt{mean}(\mathcal{E}) + 2 \cdot \texttt{std}(\mathcal{E}) \tag{3.5}$$

So that we can get a thresholded black and white edge map

$$\mathcal{E}_\tau(x, y) = \begin{cases} 1 & \text{if} \quad \mathcal{E}(x, y) > \tau \\ 0 & \text{otherwise} \end{cases} \tag{3.6}$$

Figure 8 demonstrates the process of finding closed regions from edge maps: after an edge map is thresholded, the following morphological builtin matlab operations are applied in order:

$$\begin{array}{lll} (1) & \mathcal{E}_\tau = \texttt{imdilate}(\mathcal{E}_\tau) & \textit{dilate blobs to fill gaps} \\ (2) & \mathcal{E}_\tau = \texttt{imclose}(\mathcal{E}_\tau) & \textit{close small gaps with thin lines} \\ (3) & \mathcal{E}_\tau = \texttt{imfill}(\mathcal{E}_\tau) & \textit{fill holes to} \end{array} \tag{3.7}$$

This the "thresholded" image into the "filled" image. Finally, blobs are removed if their area is less than the largest dimension of the image.

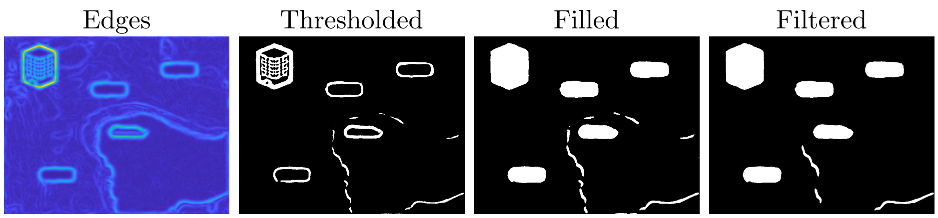

Fig. 8: The **leftmost** image shows the edges computed on an image where the added objects are scaled down by 0.75. The steps of morphological filling and filtering as explained in this section are then displayed left to right.

This process illustrates how, given corrupted Fourier samples that we would otherwise struggle to meaningfully digitize, we can extract and isolate regions of importance in the image.

3.3. Edge Informed Feature Extraction. In this thesis we are interested in how edges can inform feature extraction in images collected from Fourier samples, such as SAR or MRI images. We have three ideas to utilize edge information in feature extraction.

(1) **Regions as features:** If, for purposes such as radar target location, it suffices to identify regions of interest in a SAR image, it may very well suffice to define these regions themselves as our image features. Each feature, then, is defined as the regions center point and its surrounding shape.
(2) **Keypoints on edges:** For other purposes, such as SAR Coherent Change Detection (CCD) or SAR image registration, we may be interested in feature descriptors that maintain rotational and scale invariance, such as SIFT keypoints. However, when data are missing, it is up to experimentation, then, to determine how a SIFT feature extraction performs on an image's edge map.
(3) **Keypoints on regions*:** Our final (bonus) approach is to filter our keypoint finder's output to keypoints located inside regions so that our feature's erroneous behavior is, in a way, tamed by edge information.

In the next section, we perform numerical experiments to evaluate the efficacy of Edge Informed Feature Extraction. It is henceforth our goal to extract features that highlight the objects added to the image and ignore the background. As in (1), we may simply want to isolate these objects as regions of interests, and in (2) we attempt to coerce SIFT to focus on the objects identified by their edges. Owing to time constraints and approach (3)'s resembling a trivial masking of existing SIFT keypoints by the already-identified regions of interest, offering little novelty, we have deferred its numerical evaluation to future work.*

4. Numerical Experiments. We now present numerical experiments to assess the robustness of our feature localization pipeline under realistic synthetic aperture radar (SAR) imaging conditions. The algorithm relies on a spectral edge-detection algorithm based on the concentration factor framework developed in [3]. This technique concentrates Fourier spectral information to sharply delineate image discontinuities, yielding an effective detector of edges that accurately recovers their locations [3]. We evaluate the pipeline on several common SAR data degradation modes, including sever additive noise (low signal-to-noise ratio), missing/incomplete data, directional undersampling or occlusion simulated by radial line removal, and spatially bounded low intensity features. Each of these corruption modes presents a meaningful challenge for edge-based feature extraction. For example, heavy speckle noise can obscure true image edges and introduce spurious high-frequency artifacts [22, 4]. Likewise, missing data or limited angular converage (e.g. removal of radial lines) produces imaging artifacts such as ghosting and anisotropic blurring that distort edge features and reduce effective resolution [2]. Similarly, weak-contrast features may fail to produce discernible edges above the background, as low-magnitude edge signals are often indistinguishable from noise fluctuations [22]. This is especially problematic for our method, which identifies features as the centroids of contiguous regions extracted from the edge map; any disruption of edge continuity or clarity can therefore lead to missed detections or false feature identifications.

4.1. Tuning Edge Response. First, however, given our edge isolation method in Subsection 2.6 relies on applying a concentration factor at varying rotation angles θ_j, it will be useful to determine how many angles are required to obtain meaningful results, noting that the ideal number will ultimately depend upon data resolution, SNR, and scale contrast of the underlying image. In this regard we consider a test problem consisting of a blank image with straight, diagonal, and curved features at different magnitudes displayed in Figure 9. First, we convert the pixelized image into

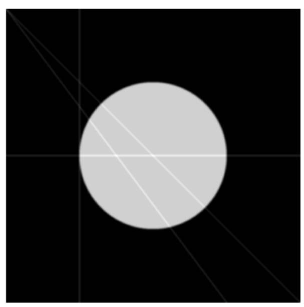

Fig. 9: One vertical, one horizontal, and two diagonal lines are overlaid onto a black image with a circle in the center. Each object is initialized with magnitude 1, and where shapes intersect the magnitues are additive. Finally, a gaussian blur with standard deviation $\sigma = 3$ is applied for smoothing.

Fourier data using (2.8). Then, to evaluate an ideal number J of angles at which to detect edge response from concentration factors, we look at the average magnitude of the edge output as well as the edge maps themselves. Each angle is given by (2.29).

Figure 10 shows that as J grows the average edge magnitude appears to converge, dipping for odd J and peaking at even J. For $J = 1$, $J = 3$, and $J = 5$ we can observe varying magnitudes in the edges around the circle with gaps at the top and on the sides. We suspect this phenomenon occurs because when J is even we evaluate θ at and $\pi/2$, but we do not when J. Since we want the strongest and most accurate possible edge response, we thus prefer $J > 10$. We can evaluate edge response in the same way with our test problem for feature extraction.

Figure 11 confirms that the edge response levels out as J grows and that $J > 10$ is a strong choice for detecting thorough and coherent edges. Although $J = 1$ and $J = 2$ return edge maps with greater average magnitude when processed on our radar image, the far left edge map in Figure 11 demonstrates the erroneous nature of a unidirectional edge response. Based on this experiment, when using the edge detection method outlined in Subsection 2.6, we will choose $J = 12$.

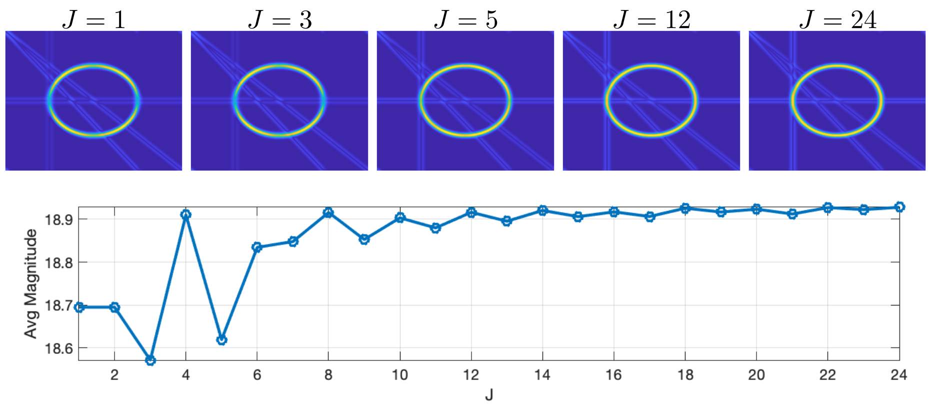

Fig. 10: Caption

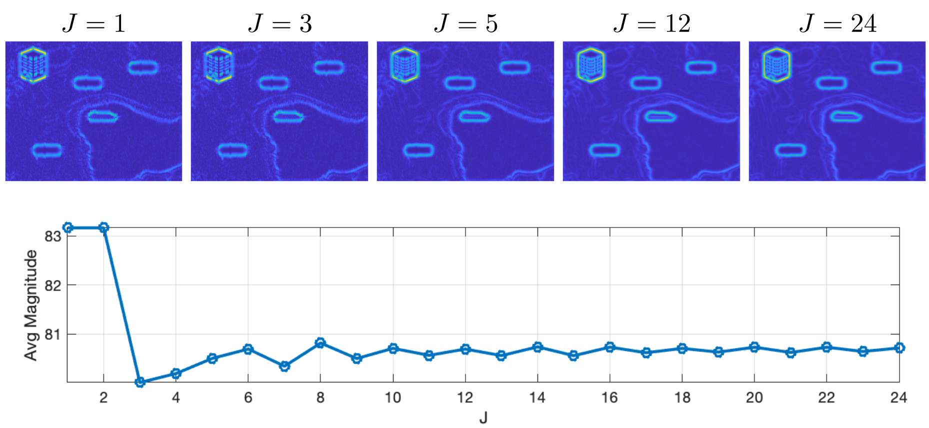

Fig. 11: Caption

4.2. Robustness to noise. Existing edge detectors and feature extraction algorithms are vulnerable to noisy Fourier coefficients. To evaluate our method's robustness under degraded conditions, noise is introduced into the test images through a physically motivated procedure. Starting from a synthetic spatial image (a toy example approximating a SAR reflectivity map), the discrete Fourier transform (DFT) (2.8) is used to obtain $F \in \mathbb{C}^{N\times N}$ with entries $F(k,l)$, $-\frac{N}{2} \le k,l \le \frac{N}{2} - 1$. Zero-mean indepedent circularly symmetric complex Gaussian noise $\hat{\boldsymbol{\epsilon}} \sim \mathcal{CN}(0, \sigma^2 I)$ is then added to obtain

$$F_{\text{noisy}}(k,l) = F(k,l) + \hat{\epsilon}(k,l), \tag{4.1}$$

where σ^2 is determined by the signal to noise ratio (SNR) which we define as

$$\text{SNR} = 10\log_{10}\left(\frac{\|F\|_2^2}{N^2\sigma^2}\right). \tag{4.2}$$

The noisy data F_{noisy} is then fed into our feature localization algorithm. This approach mirrors how SAR noise arises in practice: speckle is generated by coherent processing in the measurement (frequency) domain rather than by simple additive noise in the spatial domain [15]. The noise level is controlled by a target signal-to-noise ration (SNR, in decibels), with experiments spanning SNR values from 10 dB down to an extremely low 0.1 dB. By testing such a wide range, including very poor SNR conditions, we rigorously assess performance under increasingly severe corruption. High levels of measurement domain noise (low SNR) pose a known challenge for edge-based methods: the injected high freqency fluctuations introduce spurious edges and diminish true edge contrast [9] and if left unmitigated lead to false edge detections.

Figure 12 demonstrates our region localization method's robustness to Complex Guassian noise added in the frequency domain.

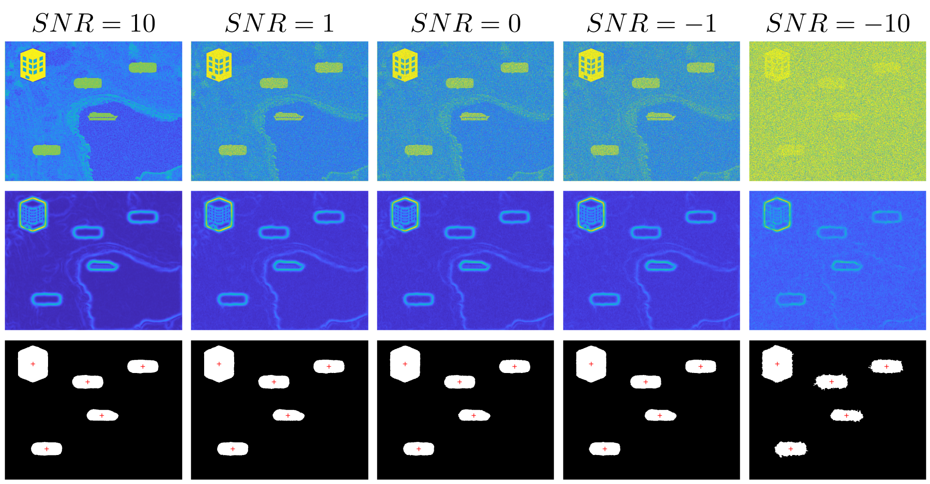

Fig. 12: In the first row, see the digitized image. In the second row, find the edge maps computed from the Fourier data. In the last row, see the isolated figures and centroids plotted as features.

Even when there is intense noise in the frequency domain, the concentration factor edge detection method outlined in Subsection 2.6 remains extremely robust to low SNR. In fact, the right two columns in Figure 12 demonstrate scenarios in which the noise is stronger than the signal itself. Where as in Subsection 2.2 spatial edge detection algorithms began to struggle at $SNR \approx 8.0$dB, our method identifies the correct edges and corresponding regions in significantly worse data conditions.

4.3. Robustness to missing data. To evaluate the resilience of our method under degraded measurement conditions, we simulate two forms of missing data in the Fourier domain. Our first approach follows the procedure outlined in Subsection 2.4 for removing radial frequencies. That is, for a given number of angles $\theta \in [0, \pi]$, we construct radial masks based on directional projections in Fourier space and zero out coefficients lying within a narrow band around each direction. This emulates directionally biased measureent loss such as occlusion, limited aperture coverage, or missing look angles.

Our second approach follows the procedure outlined in Subsection 2.4 to simulate stochastic or probabilistic data loss which can arise in SAR imagine due to irregular sampling patterns, sensor dropouts, or hardware limitations.

Both forms of corruption pose a significant challenge for edge based methods since edge detection relies on jump information and such information is spectrally distributed. Missing frequency components distort the reconstruction of discontinuities and induce anisotropic or aliased artifacts. The spectral edge detection technique employed our algorithm [9][23] relies on integrity of directional frequency content to recover edge magnitudes.

Figure 13 represents random missing frequencies (2.15) applied at varying threshold levels ρ to evaluate our method's robustness to missing data.

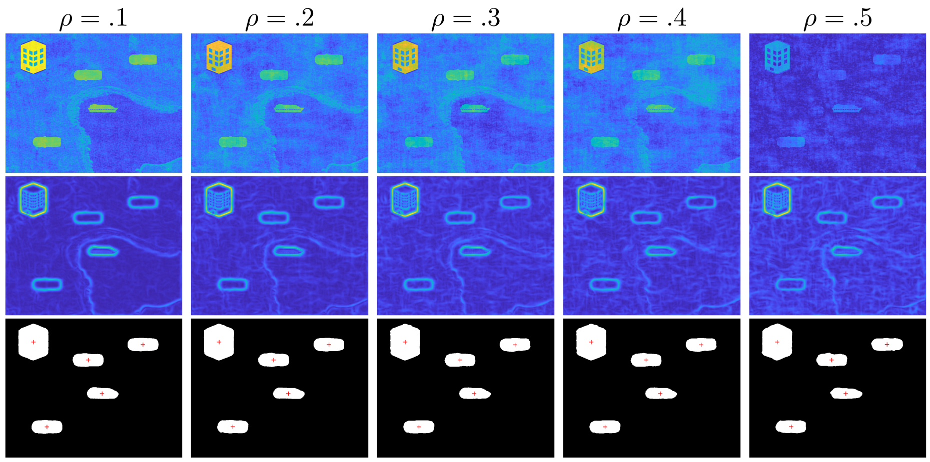

Fig. 13: In the first row, see the digitized image. In the second row, find the edge maps computed from the Fourier data. In the last row, see the isolated figures and centroids plotted as features.

Figure 13 demonstrates our methods robustness to missing data. Even with half the data missing, as shown in the fifth column, our method successfully identifies the regions of interest.

In Figure 14, we vary the number of radial lines removed from our Fourier samples to simulate another potential type of data loss. The radial lines are removed as outlined in (2.12) where J denotes the number of evenly spaced radial lines removed.

It is evident in Figure 14 that missing radial lines in sampled Fourier image data is problematic for our edge detector. As the number if missing radial lines J grows, the small gaps in the edges even after morphological operations presents a challenge for accurate region identification.

4.4. Robustness to dimness. In this experiment, we evaluated the feature algorithm under conditions where the target objects have uniformly low intensity. At low magnitudes the regions appear subtle, making them difficult to distinguish even visually in the digitized image. Such low contrast targets pose a significant challenge for edge-based feature detectors because their boundaries have weak intensity gradients. As a result, recovering these boundary jumps via spectral edge detection becomes much more difficult than in high-contrast settings. This test examines the algorithm's ability to detect subtle yet spatially significant features. This ability is important because in SAR imagine even small differences in reflectivity can reveal meaningful physical structures.

In Figure 15 we vary the scale of the objects added to the image. The objects

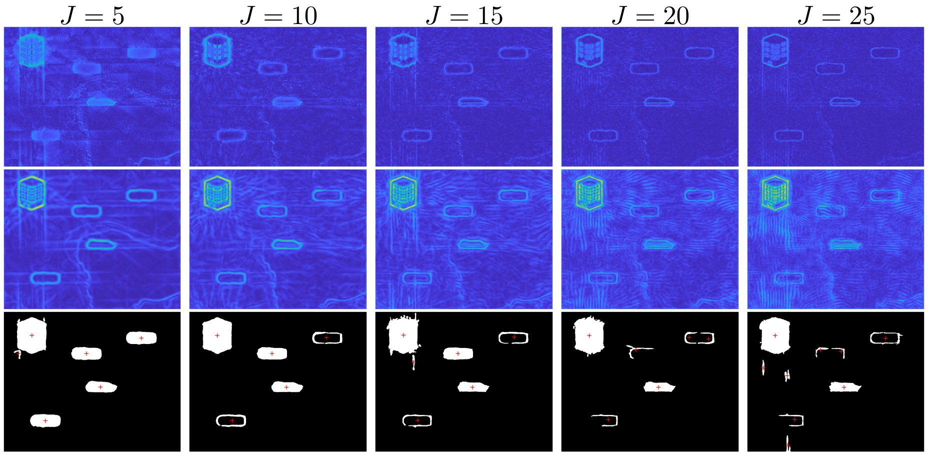

Fig. 14: In the first row, see the digitized image. In the second row, find the edge maps computed from the Fourier data. In the last row, see the isolated figures and centroids plotted as features.

are scaled by some $s \in (0, 1]$ of their pixel magnitude before being placed on top of the digitized SAR image and before being translated into the frequency domain. We assume this Fourier data is given, and proceed.

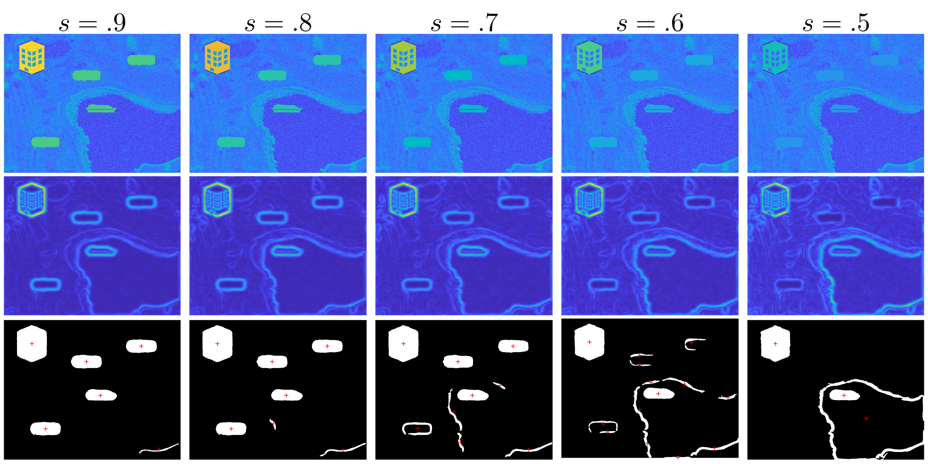

Fig. 15: In the first row, see the digitized image. In the second row, find the edge maps computed from the Fourier data. In the last row, see the isolated figures and centroids plotted as features.

Figure 15 demonstrates that objects which do not vary in brightness/reflectivity from their surroundings become significantly more difficult to pick up via edge detection. This matches our expectations given that edge detection fundamentally relies on gradient jumps.

4.5. Keypoint Identification on Edge Maps. As explained in Subsection 3.3, we are interested in extracting features from SAR images, and one way we can do that is by running an image through a keypoint identifiers such as SIFT. However, SIFT does not work on Fourier data; therefore, we must digitize the image via an inverse Fourier transform before we can extract any meaningful features.

Unfortunately, as seen in Subsection 2.1, apply standard keypoint identification algorithms on digitized images can be fruitless when the Fourier samples are plagued by strong noise or data loss. However, we know we can find the edges of images, even ones suffering such corruption.

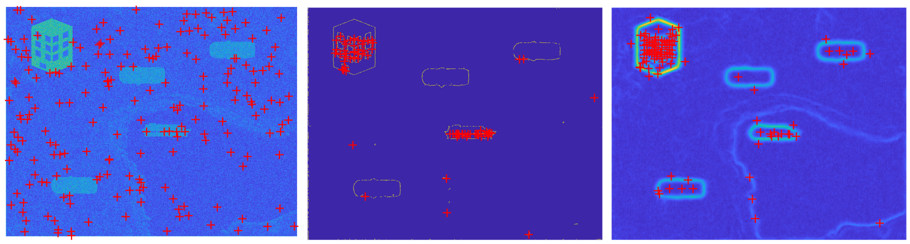

Fig. 16: 150 SIFT keypoints are plotted on (left) an image with $SNR = -1$, on (middle) the corresponding canny edge map, and on (right) the corresponding Fourier edge map.

Figure 16 demonstrates that SIFT underperforms on highly noisy images. Fortunately, using edge maps to isolate high contrast regions and suppress noisy high frequency coefficients enables SIFT to bypass excess image noise.

The same experiment can be performed for other forms of data corruption. For example, if frequencies are randomly dropped from the Fourier samples as in (2.15), we may be curious whether edge maps can serve as an adequate guide for algorithms like SIFT. In Figure 17, images, corrupted to varying degrees ρ denoting the likelihood of frequency dropout, are processed by SIFT and the corresponding keypoints are plotted over the image.

$\rho = 0.3$

$\rho = 0.5$

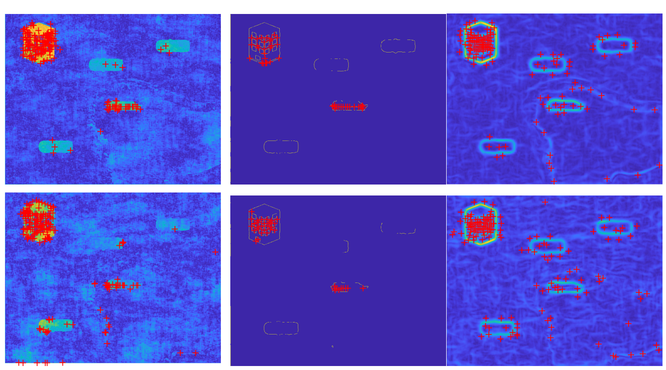

Fig. 17: 150 SIFT keypoints are plotted on (left) an image with corrupted by randomly dropped data, on (middle) the corresponding canny edge map, and (right) the corresponding Fourier edge map.

Figure 17 is shown in this way to demonstrate that the Fourier edges exhibit stronger performance than the directly digitized image at higher dropout rates in the Fourier data sample. This is expected because in Subsection 4.3, the Fourier edge detection method demonstrated robustness high random data dropout rates ρ that render the digitized image visually useless.

5. Concluding Remarks. In this work, we developed a new technique that incorporates edge information into a feature extraction pipeline. Whereas "standard" feature extraction algorithms are often designed to perform on spatially defined images, we assume our data are given as potentially corrupted or incomplete Fourier samples. There are thus significant challenges downstream when attempting to derive meaningful feature extractions out of images digitized from corrupt frequency information. Our methods combine a concentration factor based, multi-directional edge detection algorithm with morphological closing operations to isolate regions of interest that may not be identifiable from the digitized image. When data are missing, edge information may still be recoverable, so incorporating edge information into feature extraction is essential for isolating significant features from ones distracted by data corruption.

Our numerical experiments demonstrate that our new edge informed feature extraction algorithm yields improved accuracy both in isolating regions of interest in the frequency domain and in coercing traditional feature extraction algorithms to focalize on relevant areas rather than on noise. A main advantage of our approach is that we are able to overcome significant data loss by directly using the given Fourier data to determine the edge masks needed both for region isolation and for algorithm coersion.

Our initial investigations indicate that it is beneficial to use the Fourier based concentration factor method. For future work, it will be instructive to attempt this method on more challenging datasets. In addition, more attention can be paid to the morphological closing operations to overcome small gaps that are inevitable with significant data corruption, especially radial line dropount in the frequency domain.

Acknowledgments. I would like to thank Professor Anne Gelb for her encouragement and faith throughout my research and writing. It has been a joy to work alongside her on this thesis. I am so glad I took Math 76 this past fall, as I may otherwise have never discovered a curiosity for computational inverse problems. I join the guild with pride.

I must also express my gratitude to Aditya Viswanathan whose expertise was simply vital.

Lastly, I thank my family, who got me here, and my friends, whose desks, kitchen tables, and couches I often prefer to my own.

REFERENCES

[1] D. C. Aditya Viswanathan, Anne Gelb, *Iterative design of concentration factors for jump detection*, (2011).

[2] V. O. Alekseychuk, A. Kupsch, D. Plotzki, C. Bellon, and G. Bruno, *Simulation-Assisted Augmentation of Missing Wedge and Region-of-Interest Computed Tomography Data*, Journal of Imaging, 10 (2024), https://doi.org/10.3390/jimaging10010011.

[3] E. T. Anne Gelb, *Detection of edges in spectral data*, Applied and Computational Harmonic Analysis, (1999).

[4] H. Choi and J. Jeong, *Speckle Noise Reduction Technique for SAR Images Using Statistical Characteristics of Speckle Noise and Discrete Wavelet Transform*, Remote Sensing, 11 (2019), https://doi.org/10.3390/rs11101184.

[5] T. C. Ellsworth, M., *A fast algorithm for image debulrring with total variation regularization.*, Unmanned Tech Solutions, 4 (2014).

[6] K. K. Ethan Rublee, Vincent Rabaud and G. Bradski, *Orb: an efficient alternative to sift or surf*, in Proceedings of the IEEE International Conference on Computer Vision, 2011.

[7] Y. X. X. Fang Chen, *High-speed and robust scene matching algorithm based on orb for sar/ins integrated navigation system*, Scientific.net, (2012).

[8] Y. G. J. M. F. T. Flora Dellinger, Julie Delon, *Sar-sift: A sift-like algorithm for sar images*, IEEE, (2015).

[9] A. Gelb and T. Hines, *Detection of Edges from Nonuniform Fourier Data*, Journal of Fourier Analysis and Applications, 17 (2011), https://doi.org/10.1007/s00041-011-9172-7.

[10] A. Gelb and E. Tadmor, *Detection of edges in spectral data*, Applied and Computational Harmonic Analysis, 7 (1999), pp. 101–135, https://doi.org/10.1006/acha.1999.0262.

[11] T. T. H. Bay and L. V. Gool, *Surf: Speeded up robust feature*, European Conference on Computer Vision, (2006).

[12] A. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, 1989.

[13] S. L. Kang and W. K. Lee, *Sar image change detection using surf algorithm*, in Proceedings of the Third International Conference on Telecommunications and Remote Sensing - Volume 1: ICTRS,, INSTICC, SciTePress, 2014, pp. 68–73, https://doi.org/10.5220/0005421400680073.

[14] D. G. Lowe, *Distinctive image features from scale-invariant keypoints*, International Journal of Computer Vision, (2004).

[15] N. P. Manoj Singh Parihar, *Sar image noise and noise removal filter technique*, International Research Journal of Engineering Applied Sciences, (2015).

[16] L. Nguyen, *Recovery of missing spectral information in ultra-wideband synthetic aperture radar (sar) data*, (2012).

[17] U. of Edinborough Image Processing Learning Resources (online), https://homepages.inf.ed.ac.uk/rbf/HIPR2/roberts.htm.

[18] U. of Edinborough Image Processing Learning Resources (online), https://homepages.inf.ed.ac.uk/rbf/HIPR2/sobel.htm.

[19] OpenCV, *Canny edge detecton*, https://docs.opencv.org/4.x/da/d22/tutorial_py_canny.html.

[20] OpenCV, *Orb (oriented fast and rotated brief)*, https://docs.opencv.org/4.x/d1/d89/tutorial_py_orb.html#:~:text=It%20computes%20the%20intensity%20weighted,is%20the.

[21] G. Shrivakshan, *A comparison of various edge detection techniques used in image procecessing*, International Journal of Computer Science Issues, (2012).

[22] Y. Xiang, F. Wang, L. Wan, and H. You, *SAR-PC: Edge Detection in SAR Images via an Advanced Phase Congruency Model*, Remote Sensing, 9 (2017), https://doi.org/10.3390/rs9030209.

[23] A. G. G. S. Yao Xiao, Jan Glaubitz, *Sequential image recovery from noisy and under sampled fourier data*, Journal of Scientific Computing, (2022).