

ENTROPY-BASED AUTOFOCUS
FOR SYNTHETIC APERTURE RADAR

BY

ROBERT LEE MORRISON, JR.

B.S., University of Iowa, 2000

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2002

Urbana, Illinois

**ENTROPY-BASED AUTOFOCUS
FOR SYNTHETIC APERTURE RADAR**

Approved by
Supervising Committee:

To my mother and father.

ACKNOWLEDGMENTS

I would like to thank my adviser, Professor David C. Munson, Jr., for his guidance and support over the past 2 years, without which this thesis would not have been possible. He has been a compassionate mentor in times when I have felt discouraged. I express my gratitude to him for taking me on as a student, and for introducing me to this field of research. I look forward to working with him in the completion of my Ph.D.

I would also like to thank my friend and colleague Jeffrey Brokish for his advice and assistance to this work.

A special acknowledgment goes out to Dr. Charles Jakowatz, Jr., of Sandia National Laboratories for supplying the real SAR images presented in this thesis.

Lastly, I thank my parents, Dr. Robert L. and Loral Morrison, for their love and continuing support.

TABLE OF CONTENTS

CHAPTER	PAGE
1 INTRODUCTION	1
2 BACKGROUND	3
2.1 The Autofocus Problem	3
2.1.1 The inverse filtering autofocus approach	5
2.1.2 Subaperture-based approaches	7
2.1.3 Cost-based approaches	8
2.2 PGA: The Standard in Autofocus	10
2.3 SSA: A New Approach to Autofocus	14
3 EVALUATING THE MERIT OF SSA	19
3.1 Comparison of SSA with PGA: Simulated Data	19
3.2 Comparison of SSA with PGA: Actual Data	23
3.3 Efficient Implementation of SSA	25
3.3.1 Eliminating FFTs in SSA	25
3.3.2 Elimination of normalization in the entropy calculation	26
3.4 Implementation Considerations	28
3.4.1 PGA implementation	28
3.4.2 SSA implementation	29
4 EXPERIMENTS WITH SSA	45
4.1 Modifications for Efficient Computation	45
4.1.1 Multiresolution SSA	46
4.1.2 Hybrid PGA-SSA	47
4.1.3 Alternative cost functions	47
4.2 Variations of the SSA Search Strategy	49
4.2.1 Deterministic algorithms	51
4.2.1.1 Multidimensional SSA	51
4.2.1.2 Variable step size SSA	52
4.2.1.3 Perturbation round SSA	52
4.2.2 Simulated annealing algorithm	53

5 CONCLUSION	63
REFERENCES	66

LIST OF TABLES

Table	Page
3.1 Target Detection Statistics (Target Threshold = 200)	30

LIST OF FIGURES

Figure	Page
2.1 The steps of PGA demonstrated on a synthetic SAR image, at the end of the second iteration: (a) original image, (b) original image corrupted by a quadratic phase error, (c) center shifted image (after the second iteration), (d) noncoherent average $s(x)$ (the dotted line indicates the threshold—set 8.5 dB below the peak value—by which the window width is established), (e) windowed center shifted image, and (f) corrected image after the second iteration.	17
2.2 Flowchart of the SSA algorithm.	18
3.1 Synthesized test image (clutter range 0.30)—no random phase added in the spatial domain: (a) original image, (b) corrupted image, (c) PGA reconstruction (5 iterations), and (d) SSA reconstruction.	31
3.2 Blurring kernel arising from applying a white phase error, uniformly distributed on $-\pi$ and π , in the Fourier domain. The kernel is 64 elements in length. (a) Magnitude of the blurring kernel, Rayleigh distributed in the limit; (b) Phase of the blurring kernel, which is uniformly distributed between $-\pi$ and π in the limit; (c) Real part of the blurring kernel, Gaussian distributed in the limit; and (d) Imaginary part of the blurring kernel, Gaussian distributed in the limit.	32
3.3 Synthesized test image (clutter range 0.30) with random phase added in the spatial domain: (a) original image, (b) corrupted image, (c) PGA reconstruction (4 iterations), and (d) SSA reconstruction.	33
3.4 Synthesized Gaussian hill image (clutter range 0.30)—no random phase in the spatial domain: (a) original image, (b) corrupted image, (c) PGA reconstruction (2 iterations), and (d) SSA reconstruction.	34
3.5 Synthesized Gaussian hill image (clutter range 0.30) with random phase in the spatial domain: (a) original image, (b) corrupted image, (c) PGA reconstruction (4 iterations), and (d) SSA reconstruction.	35
3.6 Synthesized point target image (clutter variance 0.0052) with random phase in the spatial domain: (a) original image, (b) corrupted image, (c) PGA reconstruction (7 iterations), and (d) SSA reconstruction.	36

3.7	A plot of the quality measure, the average of the magnitudes of the 14 point targets in the reconstructed images of Figure 3.6, versus the input clutter variance, for both PGA and SSA.	37
3.8	Photographic image. The corrupted image was produced by applying a white phase error, uniformly distributed on $(-\pi$ and $\pi)$, to the Fourier data of the original image: (a) original image, (b) corrupted image, (c) PGA reconstruction (7 iterations), and (d) SSA reconstruction.	38
3.9	Photographic image of Figure 3.8 with random phase added in the spatial domain: (a) original image, (b) corrupted image, (c) PGA reconstruction (7 iterations), and (d) SSA reconstruction.	39
3.10	Defocused SAR image.	40
3.11	SSA reconstruction of the corrupt image in Figure 3.10.	40
3.12	The uncorrupted SAR image of a grassy scene.	41
3.13	The corrupted image of the grassy scene.	41
3.14	PGA reconstruction of the grassy scene.	42
3.15	SSA reconstruction of the grassy scene.	42
3.16	Enlarged portion of grassy SAR image (x coordinates 384 through 512, and y coordinates 222 through 350), (a) original image, (b) corrupted image, (c) PGA reconstruction, and (d) SSA reconstruction.	43
3.17	Plots of probability of detection vs probability of false alarm for targets in the grassy SAR image, for both PGA and SSA.	44
4.1	Multiresolution SSA (clutter range of 0.15, random phase in the spatial domain, initial downsampling factor of 4): (a) downsampled by 4, (b) downsampled by 2, and (c) initial corrupt image (left) and final result of multiresolution SSA (right).	55
4.2	Comparison of cost functions on a synthesized terrain image, with a clutter range of 0.25 and random phase applied in the spatial domain: (a) original image, (b) corrupt image, (c) SSA with entropy cost, (d) SSA with intensity-squared cost, and (e) SSA with maximum amplitude cost.	56
4.3	Comparison of cost functions on a synthesized point-target image, with a clutter range of 0.25 and random phase in the spatial domain: (a) original image, (b) corrupt image, (c) SSA with entropy cost, (d) SSA with intensity-squared cost, and (e) SSA with maximum amplitude cost.	57
4.4	Synthesized point-target image, with a clutter range of 0.35 and random phase in the spatial domain: (a) original image, (b) corrupted image, (c) entropy cost, and (d) intensity-squared cost.	58
4.5	Synthesized test image with a clutter range 0.35 and random phase in the spatial domain: (a) original image, (b) corrupted image, (c) phase error estimate is the complex conjugate of the true phase error quantized to π , (d) phase error estimate is the complex conjugate of the true phase error quantized to $\frac{\pi}{2}$, and (e) standard SSA reconstruction.	59

4.6	Entropy is not always a reliable measure of image quality. Entropy costs are expressed below, but they do not take into account the image normalization: (a) original image, with a cost of 4.57×10^2 , (b) SSA reconstruction, with a cost of 4.60×10^2 , and (c) image corrected with $\frac{\pi}{2}$ quantized phase error, with a cost of 4.69×10^2	60
4.7	Synthesized test image (clutter range 0.35 with random phase in the spatial domain): (a) original image, (b) corrupt image, (c) standard SSA, (d) multi-dimensional SSA, (e) variable step size SSA, and (f) perturbation round SSA.	61
4.8	Synthesized test image (clutter range 0.35 with random phase in the spatial domain): (a) original image, (b) corrupted image, (c) standard SSA, and (d) modified simulated annealing.	62

LIST OF ABBREVIATIONS

DFT: Discrete Fourier transform

FFT: Fast Fourier transform

ML: Maximum likelihood

PGA: Phase gradient autofocus

pdf: Probability density function

pmf: Probability mass function

SAR: Synthetic aperture radar

SSA: Stage-by-stage approaching

CHAPTER 1

INTRODUCTION

In synthetic aperture radar (SAR) imaging, phase errors occur in the demodulator due to range inaccuracies. The effect of these phase errors is to blur the radar image in the cross-range dimension. Autofocus algorithms are image restoration techniques that either directly or indirectly create an estimate of the phase error function to correct such blurred images.

There has been much work in the development of autofocus techniques in the past 25 years. One class of algorithms, inverse filtering techniques, compensates the phase error by approximating image features as ideal point functions, where the surrounding regions in the blurred image are thought to contain copies of the point spread function of the blur [1]. A widely utilized autofocus method derived from this class is phase gradient autofocus (PGA), first published by Eichel et al. in 1989 [2]. A more recent class encompasses algorithms that determine the phase error estimate that minimizes a particular cost function. The cost function is selected to measure qualities associated with focused images, such as sharpness. Several authors have proposed cost functions, but with few details regarding search strategies to achieve the minimum state. Researchers often assume a quadratic phase error, obtainable only in specific circumstances, and seek to estimate the coefficients of a quadratic function [3], [4]. A more robust method, which does not assume a particular form of the phase error, was published in a paper by Xi et al. in 1999 [5]. Their algorithm, stage-by-stage approaching (SSA), determines the estimate of the phase error function that minimizes the entropy of an image. SSA utilizes a

simple search technique to determine the entropy minimum. The authors of SSA provide examples that qualitatively demonstrate the performance of their algorithm, but they do not give a quantitative performance analysis, nor do they compare their method with other established autofocus approaches. Thus, the merits of the algorithm remain largely undetermined.

This thesis is concerned with the performance analysis and implementation of the SSA entropy-based autofocus technique. The performance of SSA is compared with that of the state-of-the-art PGA. This comparison, made quantitatively using both simulated data, and real SAR data provided by Sandia National Laboratories, reveals that SSA works extremely well. However, SSA is more computationally demanding. As a consequence, expediting implementation details are explored. In addition, novel variations of the SSA algorithm are presented. The SSA algorithm is seen to get trapped in local minima, and thus, a probabilistic searching technique is developed to increase the likelihood of achieving the global minimum entropy. Modifications aimed at improving performance and reducing computational expense are also considered.

Chapter 2 provides background information regarding the autofocus problem, and detailed descriptions of PGA and SSA. In Chapter 3, the performance of SSA is evaluated through comparison with PGA. Efficient implementation of SSA is discussed. Chapter 4 examines variations of SSA.

CHAPTER 2

BACKGROUND

2.1 The Autofocus Problem

Airborne synthetic aperture radar (SAR) produces high-resolution images by illuminating terrain with a series of pulses at various look angles. The received pulses are modulated by the complex reflectivity of the ground patch. The demodulated return is effectively the continuous-time Fourier transform of a line integral, or projection, of the reflectivity in the cross-range dimension [6]. By the projection-slice theorem, the Fourier transform of the projection at a particular angle θ , $P_\theta(U)$, is a “slice” of the 2-D Fourier transform of the complex reflectivity function $G_a(X, Y)$ taken at the same angle. The a subscript indicates that $G_a(X, Y)$ is a function of the continuous spatial frequency variables X and Y . This relationship is represented by

$$P_\theta(U) = G_a(U \cos \theta, U \sin \theta). \quad (2.1)$$

In this thesis, a monostatic imaging scenario is assumed, where the radar transmitter and receiver are colocated. The radar returns at each angle θ are sampled, yielding samples of the Fourier transform of the reflectivity function $G(X, Y)$ arranged in an annulus segment and oriented on a polar grid. The absence of an a subscript in $G(X, Y)$ indicates that it is a discrete function of X and Y . This convention will be followed throughout this thesis. These samples are interpolated onto a Cartesian grid, producing a 2-D matrix

of Fourier samples where the columns roughly correspond to the return from each radar pulse.

To properly demodulate radar returns, the two-way travel time from the radar platform to the imaged target must be known with a particular degree of accuracy. The travel time τ can be expressed as

$$\tau = \frac{2R_0}{c}, \quad (2.2)$$

where R_0 is the distance between the radar platform and the center of the scene being imaged, and c is the speed of light [1]. When τ is not known accurately, a phase error results in the demodulator that corrupts the imaging data. This phase error typically varies with each received radar echo. After interpolation to a Cartesian grid, the result is a phase error function $\phi_e(X)$ that varies with each column of Fourier data, where these columns are indexed by the cross-range spatial frequency variable X . The corrupted Fourier imaging samples $\tilde{G}(X, Y)$ are related to the uncorrupted samples $G(X, Y)$ by

$$\tilde{G}(X, Y) = G(X, Y)e^{j\phi_e(X)}, \quad (2.3)$$

where Y represents the range spatial frequency dimension, or the rows of the Fourier data matrix. The corrupted image in the spatial domain is represented by

$$\tilde{g}(x, y) = g(x, y) \otimes b(x), \quad (2.4)$$

where $\tilde{g}(x, y)$ and $g(x, y)$ represent the corrupted (defocused) and uncorrupted images, respectively, x and y are spatial dimensions, \otimes is the circular convolution operator, and $b(x)$ is a blurring kernel defined by

$$b(x) = DFT_X^{-1}[e^{j\phi_e(X)}], \quad (2.5)$$

where DFT_X^{-1} denotes an inverse discrete Fourier transform (IDFT) in the X frequency dimension. The effect of the phase error function is to act as an all-pass, blurring filter, defocusing the image in the x dimension. Since the blurring kernel is all-pass, by Parseval's relation, the uncorrupted and corrupted images have the same energy

$$\varepsilon = \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} |g(x, y)|^2 = \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} |\tilde{g}(x, y)|^2. \quad (2.6)$$

Autofocus algorithms are restoration techniques that correct defocused images by creating an estimate of the phase error function $\hat{\phi}_e(X)$. The phase error estimate is applied to the corrupted image to produce the corrected image $\hat{g}(x, y)$. To perform the image correction, the corrupt data is placed in *range-compressed* form, where a 1-D DFT is applied in the X dimension

$$\tilde{G}(X, y) = DFT_X[\tilde{g}(x, y)]. \quad (2.7)$$

The phase error estimate is applied to the range-compressed corrupted Fourier data to produce the reconstructed image $\hat{g}(x, y)$:

$$\hat{G}(X, y) = \tilde{G}(X, y)e^{-j\hat{\phi}(X)} = G(X, y)e^{j[\phi(X)-\hat{\phi}(X)]} \quad (2.8)$$

$$\hat{g}(x, y) = DFT_X^{-1}[\hat{G}(X, y)]. \quad (2.9)$$

Equation (2.8) indicates that if $\hat{\phi}(X)$ is equal to $\phi(X)$ for all X , the reconstructed image is identical to the uncorrupted image.

Several autofocus algorithms have been presented in the literature over the past 25 years. In general, these algorithms can be placed into three categories: inverse filtering or deconvolution techniques, subaperture approaches, and cost-based methods. These classes will be defined in the following subsections.

2.1.1 The inverse filtering autofocus approach

In Equations (2.4) and (2.5), the effect of the phase error was established as the discrete cyclic convolution of the uncorrupted image with an unknown blurring kernel. Without prior knowledge, the autofocus problem appears to be ill posed. The inverse filtering approach assumes a particular form of the input by approximating pointlike features in the corrupted image with ideal unit point spread functions, defined as

$$\delta(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{if } x \neq 0 \end{cases} \quad (2.10)$$

In the development that follows, only one row of the image, referred to as a *range line*, will be considered. This is permissible because the same phase error function acts upon every range line of the image.

The inverse filtering process begins by finding several isolated pointlike features, also referred to as *scatterers*, in the corrupted image. These features, in their uncorrupted form, must be free from surrounding clutter and other prominent features in order to well approximate the unit pulse in Equation (2.10). In addition, the inverse filtering approach works poorly if the *blur footprint*, the extent to which features are smeared, is larger than the distance between adjacent features. After several prominent features have been selected a window is applied to regions surrounding these features to eliminate extraneous clutter and adjacent features. The width of this window is typically chosen to be the observed size of the blur footprint. Assuming a windowed point scatterer exists on range line k , the k^{th} range line of the uncorrupted image can be approximated by

$$g(x, k) = \alpha\delta(x - x_0), \quad (2.11)$$

where the complex constant α is the reflectivity of the point scatterer, and x_0 is the displacement of the scatterer from the origin. Referring to Equation (2.4), and assuming, without loss of generality, that the scatterer is located at the origin, the corrupted range line is

$$\begin{aligned} \tilde{g}(x, k) &= \alpha\delta(x) \otimes DFT_X^{-1}[e^{j\phi(X)}] \\ &= \alpha DFT_X^{-1}[e^{j\phi(X)}]. \end{aligned} \quad (2.12)$$

The phase error estimate is now obtained by evaluating the phase of the DFT of the corrupt range line:

$$\begin{aligned} \hat{\phi}(X) &= \angle\{DFT_X[\tilde{g}(x, k)]\} \\ &= \angle\{\alpha e^{j\phi(X)}\} \\ &= \angle\alpha + \phi(X). \end{aligned} \quad (2.13)$$

Ideally, the phase error estimate equals the true phase error up to an additive constant. However, due to the surrounding clutter, the windowing process, and the lack of an ideal point scatterer, a residual phase term $\phi_r(X)$ results in the phase error estimate

$$\hat{\phi}(X) = \angle\alpha + \phi_r(X) + \phi(X). \quad (2.14)$$

Inverse filtering algorithms should be designed so that the residual phase term is made as small as possible.

The simple inverse filtering algorithm described above considers only one point scatterer when determining a phase error estimate. To obtain a more accurate estimate, information from several pointlike features should be utilized.

The inverse filtering technique suffers from several limitations. First, it is not always possible to find a strong and well isolated point scatterer in real SAR imagery, particularly if the image is of a rural scene and lacks man-made structures. In addition, the approach does not perform well when complicated phase error functions are considered, such as high-order functions and white phase error functions.

PGA, one of the most well-known and widely used autofocus approaches, uses the principles of inverse filtering to produce an estimate of the phase error function [1], [2], [7]. A detailed description of this algorithm is presented in Section 2.2.

2.1.2 Subaperture-based approaches

Subaperture-based autofocus algorithms rely upon the circular shifting property of the DFT. This property can be stated as

$$\begin{aligned} DFT^{-1}[X(m)e^{-j\frac{2\pi}{N}mk}] &= x(\langle n - k \rangle_N) \\ \text{for } m \in [0, N - 1] \text{ and } n \in [0, N - 1], \end{aligned} \tag{2.15}$$

where $x(n)$ is a length N sequence in the spatial domain, $X(m)$ is the DFT of $x(n)$, and $\langle \cdot \rangle_N$ is the modulo- N operator. Inspection of the left-hand term of Equation (2.15) reveals a similarity to Equation (2.3), the relationship between the corrupted and uncorrupted Fourier data. In both cases, the Fourier data is multiplied by a phase term that varies with each X . If the phase error function contains segments that are approximately linear for a large number of Fourier data, and an image is formed with only this subset of the Fourier data, the resulting image will be a circularly shifted version of the original image with a reduced resolution. The reduced resolution is a consequence

of using a smaller amount of Fourier data. The amount the image is shifted is proportional to the slope of the linear phase error segment.

If the phase error function is low order, such as a quadratic function, it can be well approximated by piecewise linear segments. *Map drift* is a subaperture autofocus technique that divides the Fourier data into several segments along the aperture (in the cross-range dimension). An image is formed with each of these segments. By determining the relative shift between the successive images, a piecewise linear approximation to the true phase error is formed. This approximation is accurate up to a linear phase term and a constant phase term. The unknown linear phase term results in the reconstructed image being circularly shifted with respect to the original image. This linear shift ambiguity is common to *all* autofocus algorithms. The constant phase term does not affect the magnitude of the reconstructed image, which is the part that is displayed. The relative shifts between subimages can be determined by finding the maximum in the correlation of the two subimages. Many subaperture approaches assume a phase error function that can be represented by a polynomial of a particular order, and seek to estimate the coefficients of the polynomial.

The map drift autofocus approach works well for quadratic phase error functions, which might arise in practice due to constant error in measuring aircraft velocity [1]. However, the approach performs poorly when higher-order phase functions are considered. To accurately approximate the phase function in these cases, the Fourier data must be broken into many segments, and as the size of the segments decreases, the resolution of the subimages diminishes, making it difficult to determine relative shifts between subimages.

2.1.3 Cost-based approaches

A final class of autofocus algorithms determines the phase error estimate that minimizes (or maximizes) a particular image cost function. The cost function is selected such that when it is minimized, the image is considered to be focused. Appropriateness of a cost function depends upon the type of image undergoing reconstruction.

Several image cost functions have been proposed for use in cost-based autofocus algorithms [5]. The first is the maximum value of the image amplitude, defined as

$$C_1(\hat{\phi}(X)) = -\max_{x,y}[\bar{I}_{\hat{\phi}(X)}(x,y)], \quad (2.16)$$

where \bar{I} is the power normalized image defined by

$$\bar{I}_{\hat{\phi}(X)}(x,y) = \frac{|g_{\hat{\phi}(X)}(x,y)|^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} |g_{\hat{\phi}(X)}(x,y)|^2}. \quad (2.17)$$

The $\hat{\phi}(X)$ subscript in Equations (2.16) and (2.17) indicates that the estimated image is formed by applying that phase error estimate. The power normalization step is not necessary, but it is presented here for conceptual clarity. A second proposed cost function is the ratio of the standard deviation to the mean of the image intensity, defined as

$$C_2(\hat{\phi}(X)) = -\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \bar{I}_{\hat{\phi}(X)}^2(x,y). \quad (2.18)$$

A third cost, the ratio of the standard deviation to the mean of the image amplitude, has also been proposed [3].

The cost function of primary interest in this thesis is image entropy. The entropy cost is defined as

$$C_H(\hat{\phi}(X)) = -\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \bar{I}_{\hat{\phi}(X)}(x,y) \ln[\bar{I}_{\hat{\phi}(X)}(x,y)]. \quad (2.19)$$

When an image is power normalized, it resembles a 2-D probability mass function (pmf). The power normalized image integrates to one, and the maximum value any pixel can assume is one, which only occurs when all other pixels are zero. This analogy with the pmf will establish entropy as a metric of sharpness. Intuitively, the sharpest image would be of a single unit pulse

$$\delta(x-x_0, y-y_0) = \begin{cases} 1 & \text{if } (x,y) = (x_0, y_0) \\ 0 & \text{if } (x,y) \neq (x_0, y_0) \end{cases} \quad (2.20)$$

where x_0 and y_0 are arbitrary shifts. Examination of Equation (2.19) reveals that the zero-valued pixels contribute zero to the sum, and that the one-valued pixel also contributes

zero to the sum. Thus the entropy of the “sharpest” image is zero. All other images have pixel values less than one, and these pixels make negative contributions to the sum. The negative sign in front of the double sum ensures that the entropy is always positive. The smoothest case arises when all the pixels have the same value, $\frac{1}{MN}$, yielding a maximum entropy of $-\ln[\frac{1}{MN}]$. Sharpness is associated with well-focused images. The effect of a phase error is to spread out the main lobe of a point target, though the overall energy of the image does not change, by Parseval. Thus, focused images tend to have more mass concentrated in peaks.

Once a suitable cost function has been selected, a searching algorithm must be employed to determine the phase error estimate that achieves the minimum state. Several autofocus papers that propose particular cost functions assume a polynomial form of the phase error, and then determine the coefficients of the polynomial [3], [4]. As with many of the subaperture techniques, this approach suffers from the same limitation of being constrained to a particular type of phase error. A more promising new cost-based method is the stage-by-stage approaching (SSA) algorithm [5]. It is implemented with a nonparametric estimation approach, and thus there is no restriction on the type of phase error. The phase error may be white, and the algorithm can still produce a successful reconstruction, as will be demonstrated in Chapters 2 and 3. SSA uses the image entropy cost function of Equation (2.19). The SSA algorithm will be presented in detail in Section 2.3.

2.2 PGA: The Standard in Autofocus

PGA is one of the most widely utilized and reliable autofocus algorithms. It was first published in 1989 by Eichel et al. [2]. PGA is derived from the inverse filtering class of autofocus. Unlike many of the subaperture and cost-based techniques, PGA is nonparametric. It is an iterative method, which allows the phase error estimate to be refined.

As was noted previously, the phase error is the same for every range line of the defocused image. Therefore, it makes sense to use information from multiple lines to create an estimate of the phase error. The most basic inverse filtering technique uses only one prominent scatterer to estimate the phase error, so the redundancy from other lines of the image is not exploited. PGA, in contrast, uses information from multiple prominent scatterers, producing superior reconstructions. The motivation for PGA came about in determining how to use information from multiple scatterers on different range lines to provide an optimal estimate of the common phase error function. An ad hoc approach would be to apply the inverse filtering procedure to several prominent scatterers, and then simply average the phase error estimates. However, this approach fails for images with low signal-to-clutter ratios, since the phase estimates from individual targets effectively become random variables uniformly distributed between $-\pi$ and π . The PGA algorithm employs a different averaging strategy, derived from statistical estimation theory, which will be described in the following paragraphs.

The first step in PGA is to circularly shift the brightest scatterer on each range line to the center of the image, in a step referred to as *center shifting*. The second step, *windowing*, involves the application of a window to the center shifted targets. The width of this window is selected using one of two approaches: *noncoherent averaging* or *progressive windowing*. Noncoherent averaging produces the best results when low-order phase errors are considered. When the phase error is of this type, a blur footprint is evident about the center of the center shifted image. A typical blur footprint, produced by a quadratic phase error, is seen in the center shifted image of Figure 2.1(c) (all figures and tables appear at the end of each chapter). The width of the window is determined by first summing the squared magnitude of the image in the range dimension to produce

$$s(x) = \sum_{k=1}^{N-1} |g(x, k)|^2. \quad (2.21)$$

A plot of $s(x)$, shown in Figure 2.1(d), reveals a plateau with a width the size of the blur footprint. As with basic inverse filtering, it is desired to remove extraneous clutter and competing targets, so the window should only be large enough to pass the blurring

artifacts resulting from a single scatterer. If the window is too small, information about the effect of the defocusing will be lost, and the phase error estimate will be inaccurate. To determine the best window width, a threshold is placed 5 to 10 dB below the peak value of $s(x)$. The edges of the window are then defined as the points where the threshold intersects $s(x)$. The window is now approximately the size of the blur footprint, as Figure 2.1(e) demonstrates. Progressive windowing, useful when higher-order or random phase errors are involved, starts with a fixed window and then decreases the window size by a particular factor with each iteration. This approach is necessary when a distinct blur footprint is not evident.

After the windowing step, the windowed image is placed into range-compressed form. The windowed image can be thought of as a column of point targets, all of which have been blurred by the same unknown phase error. Under this assumption, the range-compressed data is constant in magnitude on each range line. A model for the range-compressed data is given by

$$\bar{g}(X, y) = \alpha(y)e^{j\phi(X)} + \eta(X, y), \quad (2.22)$$

where $\alpha(y)$ is the magnitude of the point scatterer on each range line y , $\phi(X)$ is the phase error, and $\eta(X, y)$ is a noise term that models the interfering clutter surrounding a point target. $\eta(X, y)$ is assumed to be zero mean, white, Gaussian, and independent of $\alpha(y)$. If there were no phase error, the phase would also be constant on each range line. Therefore, the differences in phase between adjacent frequency samples in the X dimension provide an estimate of the phase error (up to a constant and a linear shift). This estimate will not be exact because of the noise term $\eta(X, y)$ and also because the uncorrupted scatterers are not really ideal point targets. Thus, the question becomes how to use the phase estimates from each line to form a comprehensive phase estimate. A *phase difference algorithm*, based on maximum likelihood (ML) estimation theory, has been proposed as a solution. It has been shown that the phase error estimate resulting from this approach is optimal with respect to the assumptions made about $\eta(X, y)$ [1].

The first step of the phase difference algorithm is to take the inner product of adjacent Fourier samples. The phase of this inner product is the phase difference between samples.

Intuitively, it might make sense to find the phase difference between samples, and then average down the columns to find a comprehensive phase difference. However, this is not optimal in the ML sense; the ML estimate is obtained by first accumulating the inner products along the columns, and then taking the angle. This is expressed mathematically as

$$\Delta\hat{\phi}(X) = \angle \sum_{k=0}^{N-1} \{\bar{g}^*(X-1, k)\bar{g}(X, k)\}, \quad (2.23)$$

where $\Delta\hat{\phi}(X)$ is the ML estimate for the phase error difference between adjacent Fourier samples, and the $*$ superscript denotes complex conjugation. The phase error estimate is obtained by forming a running sum of the phase differences:

$$\hat{\phi}(X) = \sum_{l=1}^X \Delta\hat{\phi}(l). \quad (2.24)$$

Once the phase error estimate has been computed, it is applied according to Equation (2.8) to form the corrected image. PGA is then reapplied to this corrected image to further focus the image. After each iteration, the energy of the phase error estimate (for that iteration) is compared with a threshold. If the energy is greater than the threshold, the algorithm is considered to have made progress, and another iteration is commissioned. Otherwise, the algorithm terminates. It is important to note that the phase error estimate produced with each iteration is only an incremental estimate, and to achieve the overall phase error estimate, the incremental estimates must be accumulated.

PGA is an innovative algorithm that offered significant improvement to previous approaches. It has become one of the standards in SAR autofocus—a claim evident through observation of the large body of published work it has influenced. However, the algorithm is limited to applications where a sufficient number of isolated pointlike structures are present. PGA can perform poorly on challenging images with low signal-to-clutter ratios or random phase errors.

2.3 SSA: A New Approach to Autofocus

The SSA autofocus algorithm was published in 1999 by Xi et al. [5]. This technique determines the phase error estimate that minimizes the *entropy* of an image. The SSA approach is promising because it is nonparametric, and because it produces successful reconstructions for white phase error functions. It does not suffer the limitations of inverse filtering or PGA, since a large number of isolated pointlike structures (free of surrounding clutter) are not required. The search strategy for obtaining the minimum entropy is conceptually straightforward, and does not require the computation of a function gradient.

The innovation of SSA lies both in its choice of a cost function, entropy, and in its searching strategy. The motivation for using entropy as a prior was presented in Section 2.1.3. To reiterate, images with lower entropy are more likely to correspond to focused images than are images with higher entropy. SSA does not assume a particular form of the phase error. If there are M columns of Fourier data, then there are M elements of the phase estimate $\hat{\phi}$ to be determined. Finding a $\hat{\phi}$ that minimizes the entropy corresponds to finding the minimum location in an M -dimensional entropy space. Searching such a complicated, high-dimensional space with conventional optimization techniques, which may involve calculation of a gradient, is far from straightforward [5]. SSA utilizes a simple search technique that searches in only one dimension at a time. Each element of $\hat{\phi}$ is perturbed by adding or subtracting a corrective factor, allowing the algorithm to take a step in the corresponding dimension in entropy space. In the process of searching for the entropy minimum, SSA decreases its searching step size by a factor of two with each iteration.

Let the phase error estimate $\hat{\phi}$ be an M by 1 vector, initialized to all zeros. Each element of $\hat{\phi}$ is perturbed, in sequence, by either adding or subtracting the corrective factor Δ , which is initialized to π at the beginning of the algorithm. After each perturbation,

three candidate phase error estimates result:

$$\begin{aligned}
\hat{\phi}_{0,n+1}(X) &= \hat{\phi}_n(X) \\
\hat{\phi}_{1,n+1}(X) &= \hat{\phi}_n(X) + \Delta\delta(X - m) \\
\hat{\phi}_{2,n+1}(X) &= \hat{\phi}_n(X) - \Delta\delta(X - m),
\end{aligned} \tag{2.25}$$

where n is the current state, $n + 1$ is the next state, and $\delta(X - m)$ is a vector that is one in the m^{th} element (the element that is perturbed) and zero elsewhere. Each candidate vector is applied to $g_{\hat{\phi}_n(X)}(X, y)$ as prescribed by Equation (2.8), producing three images. The entropy of each image is computed. The phase error estimate associated with the lowest entropy image is selected as the next state, which can be expressed as

$$\hat{\phi}_{n+1} = \underset{\hat{\phi}_{j,n+1}}{\text{arg min}} [C_H(\hat{\phi}_{j,n+1})] \quad j = 0, 1, 2, \tag{2.26}$$

where $C_H(\cdot)$ is the entropy function defined in Equation (2.19). The perturbation of all M elements constitutes one *subiteration*.

After a subiteration has commenced, the relative change in entropy is computed:

$$D_0 = \left| \frac{C_H(\hat{\phi}_0) - C_H(\hat{\phi}_{M-1})}{C_H(\hat{\phi}_0)} \right|, \tag{2.27}$$

where $\hat{\phi}_0$ is the phase estimate at the beginning of the subiteration, and $\hat{\phi}_{M-1}$ is the phase estimate after all M elements have been perturbed. If D_0 is greater than the tolerance T_0 , the subiteration has produced a significant improvement to the phase error estimate, and another subiteration is commissioned at the same step size Δ . In this case, the current $\hat{\phi}$ is perturbed in the same manner as in the previous subiteration. If D_0 is less than T_0 , the current iteration i ends, and a new one begins. The searching step length is updated with each new iteration. These events are expressed as

$$\begin{aligned}
i &= i + 1 \\
\Delta_i &= \Delta_{i-1}/2
\end{aligned} \tag{2.28}$$

The term Δ decreases by a factor of two with each iteration. In each iteration, subiterations occur with step size Δ_i until the relative change in entropy D_0 falls below T_0 .

The SSA algorithm terminates when the relative change in entropy D_1 between *iterations* falls below an outer-loop threshold T_1

$$D_1 = \left| \frac{C_H(\hat{\phi}_{i-1}) - C_H(\hat{\phi}_i)}{C_H(\hat{\phi}_{i-1})} \right|, \quad (2.29)$$

where $\hat{\phi}_i$ is the phase error estimate after the i^{th} iteration. A flowchart of the SSA algorithm is shown in Figure 2.2.

The SSA autofocus approach produces impressive reconstructions of phase-corrupted images. However, the authors of SSA made no comparison with other autofocus approaches. An objective of this thesis is to provide such a comparison. In Chapter 3, the merits of SSA are evaluated through comparison with the established PGA algorithm outlined in Section 2.2. This comparison is made using both real and simulated SAR data. Chapter 3 also explores implementation details not explicitly described in the original SSA paper, which allow for efficient execution of the algorithm. Chapter 4 suggests modifications to the SSA search strategy that reduce the likelihood of the algorithm becoming trapped in a local minimum (not finding the absolute minimum entropy state).

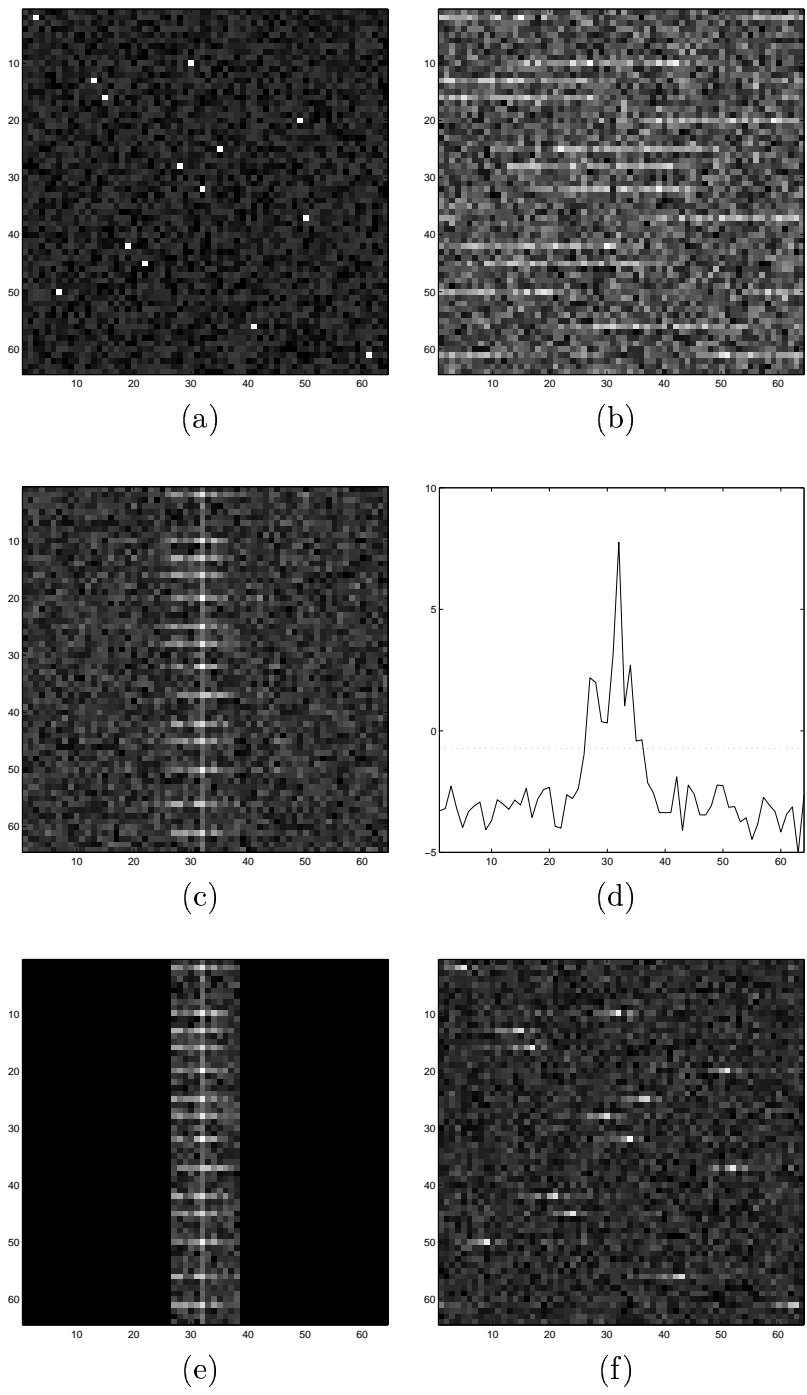


Figure 2.1 The steps of PGA demonstrated on a synthetic SAR image, at the end of the second iteration: (a) original image, (b) original image corrupted by a quadratic phase error, (c) center shifted image (after the second iteration), (d) noncoherent average $s(x)$ (the dotted line indicates the threshold—set 8.5 dB below the peak value—by which the window width is established), (e) windowed center shifted image, and (f) corrected image after the second iteration.

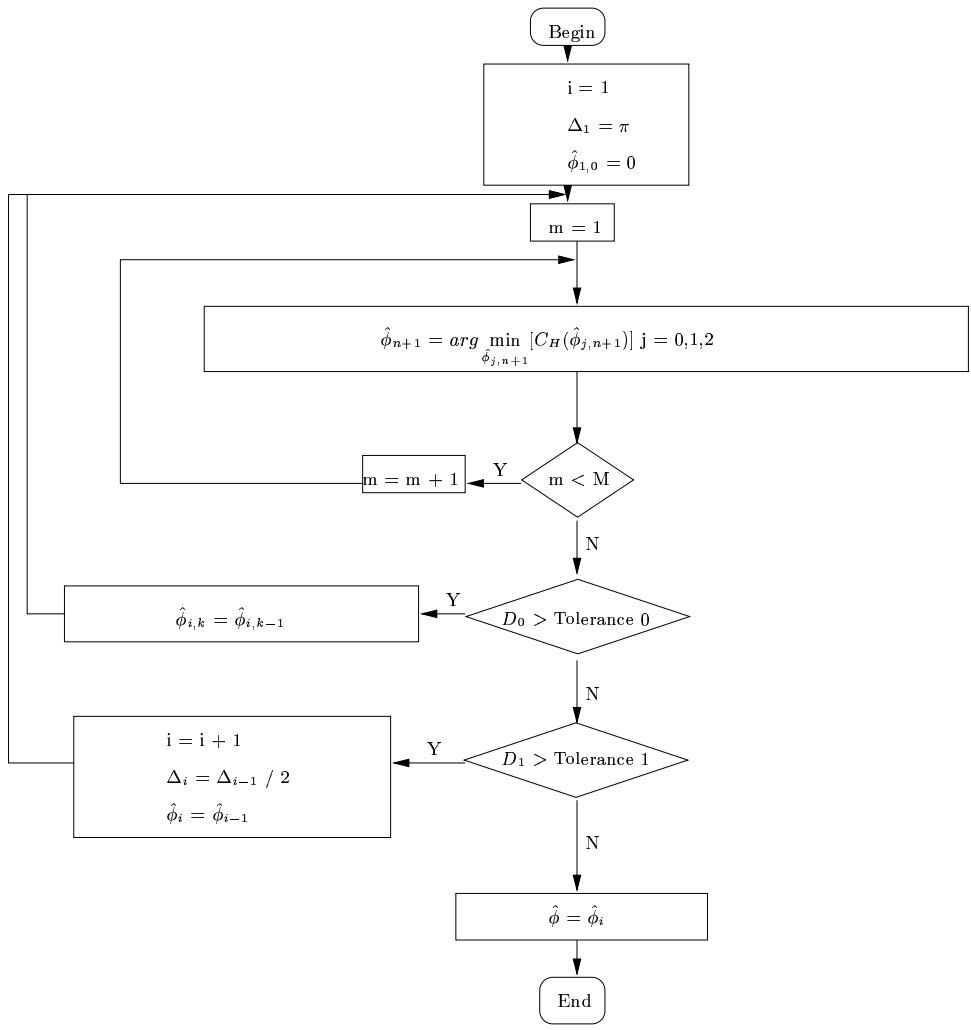


Figure 2.2 Flowchart of the SSA algorithm.

CHAPTER 3

EVALUATING THE MERIT OF SSA

3.1 Comparison of SSA with PGA: Simulated Data

To compare the performance of SSA and PGA we created synthetic images. The first contains several “buildings” and an elongated “kitelike” structure that has the appearance of a wall. These structures are referred to as *targets*. This image, shown in Figure 3.1(a), is 64 by 64 pixels. The targets in the image take on either purely real or purely imaginary values $\in [0.25, 1]$, where brighter pixels represent values closer to one. The image is formed by first creating a 64 by 64 matrix, where each pixel is assigned an independent, uniformly distributed value $\in [0, \gamma]$, where γ is a variable parameter. A γ value of 0.3 is used in this simulation. The assignment of random values to each pixel simulates the clutter present in real SAR images. The targets are then introduced to the image.

The corrupted image of Figure 3.1(b) is formed by applying a white phase error function, uniformly distributed in $-\pi$ and π and varying only in the cross-range dimension, to the range-compressed original image. A plot of the magnitude of the resulting blurring kernel is shown in Figure 3.2(a). This plot reveals several dominant pointlike impulses, such that the resulting corrupted image is a superposition of scaled and circularly shifted replicas of the real and imaginary components of the original image. As a consequence, replicas of the elongated “kitelike” structure can be seen in the corrupt image. The magnitude of the blurring kernel is Rayleigh distributed. The pointlike impulses result

because the Rayleigh probability density function (pdf) has a heavy tail that extends to infinity. The real and imaginary components of the blurring kernel, shown in Figures 3.2(c) and (d), respectively, are Gaussian distributed.

Actually, the Rayleigh and Gaussian pdfs are the *limiting* pdfs for a large image size (and therefore a long phase sequence). Consider the phase error function

$$\begin{aligned}\phi(k) &\sim U(-\pi, \pi) \\ Z(k) &= e^{j\phi(k)},\end{aligned}\tag{3.1}$$

where $U(-\pi, \pi)$ denotes a uniform distribution between $-\pi$ and π . The impulse response, $z(n)$, is determined by taking the inverse DFT of $Z(k)$, which is expressed as

$$\begin{aligned}z(n) &= DFT^{-1}[Z(k)] \\ &= \frac{1}{N} \sum_{k=0}^{N-1} Z(k) e^{j\frac{2\pi}{N}kn} \\ &= \frac{1}{N} \sum_{k=0}^{N-1} e^{j(\phi(k) + \frac{2\pi}{N}kn)} \\ &= \frac{1}{N} \sum_{k=0}^{N-1} e^{j\phi'(k,n)}\end{aligned}\tag{3.2}$$

where

$$\phi'(k, n) = \phi(k) + \frac{2\pi}{N}kn.\tag{3.3}$$

Each exponential term in Equation (3.2) is uniformly distributed around the circle in the complex plane. Therefore, $z(n)$ can be considered a sum of random phasors with unit magnitude, a quantity frequently considered in optics [8]. The real and imaginary components of Equation (3.2) are

$$\begin{aligned}z_r(n) &= \frac{1}{N} \sum_{k=0}^{N-1} \cos(\phi'(k, n)) \\ z_i(n) &= \frac{1}{N} \sum_{k=0}^{N-1} \sin(\phi'(k, n)).\end{aligned}\tag{3.4}$$

By the central limit theorem, as N approaches infinity, $z_r(n)$ and $z_i(n)$ become Gaussian distributed with mean zero and variance $\sigma_r^2 = \sigma_i^2 = \frac{1}{2N}$. Utilizing a transformation of variables, $|z(n)|$ can be shown to have a pdf approaching a Rayleigh distribution with

mean $\frac{1}{2}\sqrt{\frac{\pi}{N}}$ and variance $\sigma_{|z|}^2 = \frac{1}{2N}[2 - \frac{\pi}{2}]$ for sufficiently large N . The phase of the blurring kernel is uniformly distributed on $(-\pi, \pi)$ [8]. A plot of the phase is displayed in Figure 3.2(b).

The PGA and SSA algorithms were applied to the corrupted image in Figure 3.1(b). The resulting reconstructions are displayed in Figures 3.1(c) and (d), respectively. Initially, the reconstructions were circularly shifted with respect to the original image. This shift ambiguity, described in Section 2.1.2, is common to all autofocus algorithms. To facilitate comparison, the reconstructions were circularly shifted back in line with the original image. The PGA reconstruction exhibits some blurring, evident in the smeared appearance of the background and defocus of the targets. The SSA reconstruction reveals fewer blurring artifacts. It is nearly identical to the original image.

In Figure 3.3(a), a random phase, uniformly distributed between $-\pi$ and π and independent from pixel to pixel, was applied in the spatial domain to model the complex reflectivity of background terrain found in real SAR images. This image resembles that of Figure 3.1(a), since only the magnitude is displayed; the addition of random phase does not affect the magnitude. The corrupt image, shown in Figure 3.3(b), is now the result of a circular convolution of a complex image with a complex valued blurring kernel. The corrupt image no longer reveals shifted replicas of the kite structure. Comparison of Figures 3.1 and 3.3 reveals that the quality of the PGA reconstruction is much worse when random phase is applied to the original image. This random phase scenario more closely simulates a real SAR autofocus problem. The SSA reconstruction appears well focused. The quality of the reconstruction is not hindered by the addition of random phase in the spatial domain.

The first image contains sharp pointlike targets, a scenario in which PGA and SSA would be expected to perform well. To test the robustness of the algorithms on a smooth image, free of any point structures, an image containing two Gaussian “hills” was synthesized. This image is displayed in Figure 3.4. The PGA and SSA reconstructions are shown in Figures 3.4(c) and (d), respectively. SSA produces an accurate reconstruction of the original image. The PGA reconstruction is somewhat more defocused.

The Gaussian hill image of Figure 3.5(a) has random phase incorporated in the spatial domain. The corrupted image of Figure 3.5(b) is defocused to a much greater extent. The corrected images are displayed in Figures 3.5(c) and (d). PGA and SSA produce similar reconstructions, with the PGA result being slightly more defocused. The reconstructions maintain the general shape of the Gaussian hills, although these features are not as smooth.

Figure 3.6(a) is a synthesized image composed of 14 real-valued point targets surrounded by uniformly distributed clutter. A corrupted version of this image, formed by perturbing the Fourier data of the original image with a white phase error function, is displayed in Figure 3.6(b). Applying PGA and SSA to the corrupted image produced the reconstructions shown in Figures 3.6(c) and (d), respectively. Inspection of these images clearly shows that SSA produced a more accurate reconstruction than PGA. The point targets in the PGA image were blurred in cross-range, and several false targets appeared.

To quantitatively compare the performance of PGA and SSA, Figure 3.7 shows plot of a quality measure for both algorithms as a function of the clutter variance. A set of images is formed with the 14 point targets of Figure 3.6, with each image assigned a different clutter variance. The quality measure is defined to be the average of the magnitudes of the 14 point targets in the PGA and SSA reconstructed images. Magnitude is a useful quality measure because the effect of the phase error is to widen the main lobe of the point targets, thus necessarily decreasing their amplitudes, since the phase error preserves total image energy (Parseval's relation). Figure 3.7 shows that SSA performs consistently well over a wide range of clutter variances, but then breaks down above a certain threshold. The performance of PGA, in contrast, progressively degrades with increasing clutter variance.

Figure 3.8(a) displays a segment of a photographic image. The image was corrupted by a white phase error to produce the defocused image of Figure 3.8(b). The purpose in using a photograph was to determine if the entropy criterion for evaluating focus is appropriate for images that do not resemble SAR images. Figures 3.8(c) and (d) show the PGA and SSA reconstructions, respectively. In contrast to the previously considered

simulations, PGA produces a far superior reconstruction. The SSA corrected image offers little improvement to the corrupted image. The entropies of the original image, the PGA reconstruction, and the SSA reconstruction were computed, and the SSA image indeed has the lowest entropy. Thus, the minimum entropy criterion does not correspond to the most focused image for this example. The poor performance by SSA in this case indicates that entropy is not always an appropriate cost function, although it appears to work extremely well for SAR-like images. The effectiveness of SSA on actual SAR data is examined in Section 3.2.

In Figure 3.9(a), a random phase was applied in the spatial domain to the image of 3.8(a). The corresponding corrupt image of 3.9(b) is completely obscured and in no way resembles the original image. Figures 3.9(c) and (d) indicate that PGA and SSA could not reconstruct the corrupted image.

3.2 Comparison of SSA with PGA: Actual Data

The results outlined in the previous section illustrate the potential of SSA on simulated SAR data. The utility of SSA on real SAR data is now considered. Sandia National Laboratories, which developed the PGA algorithm, provided us with real SAR images for this purpose.

Figure 3.10 is an actual SAR image corrupted by an unknown phase error function. The image is 600 by 600 pixels, and the image data was supplied in the form of complex signed 16 bit integers. A SSA reconstruction is displayed in Figure 3.11. A fence, or the perimeter of a building, is evident in the corrected image. The cross-range blur has been reduced to reveal features that resemble cars (near coordinates (350,500)).

A second set of SAR images is of a grassy scene. In addition to a phase corrupted image, Sandia National Laboratories provided the original image and a PGA reconstruction of the corrupted image using their version of PGA. The original image is displayed in Figure 3.12, the corrupt image in Figure 3.13, and the Sandia National Laboratories PGA reconstruction in Figure 3.14. The uncorrupted image of Figure 3.12 contains

many pointlike scatterers. The full-sized (512 by 512 pixels) image displayed with an appropriate color map on a computer monitor reveals the pointlike features much more distinctly than the black and white image presented here. However, several prominent scatterers, such as the points at coordinates (234,200) and (259,198), are evident by inspection of the figure. Figure 3.15 shows a SSA reconstructed image, produced using our implementation of SSA.

We magnified an interesting portion of the grassy image to provide a basis for qualitative comparison. This excerpt, displayed in Figure 3.16, reveals rowlike features. These “rows” are defined more distinctly in the SSA reconstruction. In the PGA reconstruction the rows blur together. The targetlike features depicted in the uncorrupted image segment are obscured in the PGA image, and extraneous features abound. The SSA reconstruction, in contrast, yields a faithful restoration of prominent features.

To quantitatively compare the performance of PGA and SSA on the grassy image, we analyzed the correct and incorrect reconstructions of pointlike features. The pointlike scatterers found in these images were considered “targets.” A threshold was applied to the original, SSA-corrected, and PGA-corrected images. If the pixel value in the images was greater than the threshold, the coordinates of that pixel were noted, and that pixel was considered to be a target (if multiple adjacent pixels exceeded the threshold, only one of the pixels was considered a unique target). Five lists were constructed: targets in the original image, targets detected correctly with PGA, targets detected incorrectly with PGA (false alarms), SSA correct detections, and SSA false alarms. The detection statistics are summarized in Table 3.1.

The probability of correct detection is expressed as the conditional probability that a system detects a target given that a target is actually present. Likewise, the probability of false alarm is defined as the conditional probability that the reconstructed image indicates a target when no corresponding target exists in the original image. The true targets are considered to be those in the original image exceeding a threshold of 200. The probability of detection versus false alarm for PGA and SSA was then determined by applying 50 thresholds in the range of 185 to 235 to the reconstructed imagery. The resulting plot is

displayed in Figure 3.17. This plot indicates that the SSA algorithm achieved a superior probability of detection versus false alarm. This is notable, considering that all earlier PGA results in this thesis used our own implementation of PGA. The PGA results for the grassy scene, however, are from Sandia, using presumably the best version of PGA available.

3.3 Efficient Implementation of SSA

The previous sections illustrate the potential of SSA as a powerful and robust autofocus tool. However, the computational expense of the algorithm is a major drawback that must be addressed before SSA can be placed into widespread use. The fastest implementation of SSA still requires approximately 10 times the computational time of PGA for images of size 512 by 512 pixels. We employed two expediting measures not explicitly mentioned in the original SSA paper. The first was the replacement of complete 1-D fast Fourier transforms (FFTs) after each perturbation with a more efficient recursive image update. The second was the elimination of the normalization step in the entropy calculation. These expediting measures are discussed in the following subsections.

3.3.1 Eliminating FFTs in SSA

A detailed description of the SSA algorithm was presented in Section 2.3. The heart of the SSA algorithm lies in perturbing each phase element with three options to produce three candidate phase error estimates. The estimate that yields the lowest entropy image is selected as the next state. To determine the next state, each candidate estimate must be applied to the range-compressed data. A series of 1-D inverse FFTs must then be applied to create the image, so that the entropy function can be evaluated. For an image with 512 columns, it would seem that 1536 inverse FFTs must be computed in each subiteration. Fortunately, a full 1-D FFT need not be performed with each perturbation. It is possible to define the phase perturbed images recursively, such that the next state

image can be produced by adding an update matrix to the current state image [9]. The update equation and its proof are presented in the following paragraphs.

The range compressed data at the $(n + 1)^{th}$ state can be represented as

$$G_{n+1}(X, y) = G_0(X, y)e^{j(\phi_n(X) + \Delta_n \delta(X - K_n))} \quad (3.5)$$

where G_0 is the range compressed data before any phase perturbations have been applied, $\phi_n(X)$ represents the cumulative phase perturbations up through state n , and $\Delta_n \delta(X - K_n)$ is the perturbation applied to column K_n in the $(n + 1)^{th}$ update. The updated range-compressed data G_{n+1} can be expressed as a function of the previous state data G_n :

$$G_{n+1}(X, y) = G_0(X, y)e^{j\phi_n(X)} - G_0(K_n, y)e^{j\phi_n(K_n)}\delta(X - K_n) + G_0(K_n, y)e^{j(\phi_n(K_n) + \Delta_n)}\delta(X - K_n) \quad (3.6)$$

$$G_{n+1}(X, y) = G_n(X, y) + (e^{j\Delta_n} - 1)G_0(K_n, y)e^{j\phi_n(K_n)}\delta(X - K_n). \quad (3.7)$$

The next state image is therefore expressed as

$$I_{n+1}(x, y) = I_n(x, y) + \frac{1}{N}(e^{j\Delta_n} - 1)G_0(K_n, y)e^{j\phi_n(K_n)}e^{j\frac{2\pi}{M}K_n x}. \quad (3.8)$$

Equation (3.8) gives an expression for the $(n + 1)^{th}$ image in terms of the n^{th} image plus a term that depends on the phase perturbation associated with the update. This result is useful because it shows that if only one element of the phase function is changed, an inverse DFT does not need to be applied to obtain the modified image. In the SSA algorithm, only one phase element is perturbed at a time, so Equation (3.8) has enabled the elimination of all FFT's from the algorithm except for the initial one-dimensional FFT required to compute G_0 . The complexity of the image update has been reduced from $O(n^2 \log n)$ to $O(n^2)$.

3.3.2 Elimination of normalization in the entropy calculation

The notion of eliminating the normalization step of the the entropy calculation at first appears preposterous; the assumptions that enable an analogy between an image and a probability density function all hinge upon this step. However, it can be shown that the

image need not be normalized at the beginning of each entropy calculation as Equations (2.17) and (2.19) suggest. This calculation can be “redistributed” in such a way that the normalization effect need only be considered when computing the termination criteria of Equations (2.27) and (2.29) [9].

Let $G(X, y)$ represent the range-compressed data of an image, and let $\tilde{G}(X, y)$ represent the data perturbed by the phase function $\phi(X)$ as follows:

$$\tilde{I}(x, y) \longrightarrow \tilde{G}(X, y) = G(X, y)e^{j\phi(X)}. \quad (3.9)$$

The entropy function applied to the magnitude squared of the image is

$$\tilde{H} = - \sum_{x,y} |\tilde{I}|^2 \ln |\tilde{I}|^2. \quad (3.10)$$

The power normalized image is expressed as

$$\bar{I} = \frac{|\tilde{I}|^2}{\sum_{x,y} |\tilde{I}|^2}. \quad (3.11)$$

The entropy of the power normalized image is

$$\bar{H} = - \sum_{x,y} \bar{I} \ln \bar{I}. \quad (3.12)$$

The denominator of Equation (3.11) can be expressed as

$$\begin{aligned} \sum_{x,y} |\tilde{I}|^2 &= \frac{1}{MN} \sum_{X,Y} |G|^2 \\ &= \gamma. \end{aligned} \quad (3.13)$$

Equation (3.13) indicates that the denominator takes on a positive constant value regardless of the phase function ϕ applied. Equation (3.12) can now be expressed as

$$\begin{aligned} \bar{H} &= - \sum_{x,y} \frac{|\tilde{I}|^2}{\gamma} \ln \frac{|\tilde{I}|^2}{\gamma} \\ &= \frac{1}{\gamma} \tilde{H} + \ln \gamma, \end{aligned} \quad (3.14)$$

utilizing Equations (3.10) and (3.13). The minimum entropy is invariant under constant addition and multiplication by a positive scalar. Thus, minimizing \tilde{H} is equivalent to

minimizing \bar{H} , which is beneficial because \tilde{H} does not require a normalization step. The only thing that needs to be changed is the loop termination criteria. This change is accomplished as follows:

$$\begin{aligned} D_i &= \frac{\bar{H}_0 - \bar{H}_1}{H_1} \\ &= \frac{\tilde{H}_0 - \tilde{H}_1}{H_0 + \gamma \ln \gamma}. \end{aligned} \tag{3.15}$$

Thus, the loop termination criteria are modified by adding the term $\gamma \ln \gamma$ in the denominator.

3.4 Implementation Considerations

The following subsections describe details specific to the PGA and SSA implementations used in this chapter.

3.4.1 PGA implementation

In the implementation of PGA used in Section 3.1, a few minor modifications were made that did not alter the methodology of the algorithm. First, the energy termination criteria, which compares the energy of the incremental phase estimates to a threshold, was not found to be an effective means of terminating the algorithm. Therefore, instead of using this criterion, enough iterations were completed until the best image was obtained. Typically, there is a nominal number of iterations, below and above which the image quality decreases.

The authors of PGA suggest using the progressive windowing scheme described in Section 2.2 when random phase errors are considered. In practice, this approach was found to perform poorly because the window width reduces either too quickly or too slowly, and the nominal schedule depends on the type of image under consideration. Therefore, a variation of the noncoherent averaging scheme was employed. In contrast to the quadratic phase error case, where the noncoherent average defined in Equation (2.21) exhibits a plateau that is always greater than the windowing threshold, the noncoherent average formed in the random phase error case may cross the threshold several times.

The window in this case is defined either by the point where the noncoherent average first crosses the threshold (the leftmost edge), or by the point where it last crosses the threshold (the rightmost edge). The width of the window is defined to be twice the distance from the center of the image to the farthest edge. The windowing threshold in this implementation of PGA was selected to be 8.5 dB below the maximum value of the noncoherent average.

3.4.2 SSA implementation

The implementation of SSA used in Sections 3.1 and 3.2 employs the efficiency measures described in Sections 3.3.1 and 3.3.2.

The values of the termination thresholds were $T_0 = 1 \times 10^{-4}$ and $T_1 = 0.01 \times 10^{-4}$. Increasing T_0 causes the algorithm to proceed to the next iteration more quickly. However, this often results in the algorithm terminating prematurely, or poor image reconstructions. Increasing T_1 decreases the number of iterations SSA performs. Since many images can be corrected reasonably well with coarse step sizes, using a large value of T_1 may be an effective way to achieve a faster implementation with only minor image degradation.

In Chapter 4, variations of the SSA algorithm aimed at improving performance and computational expense are explored.

Table 3.1 Target Detection Statistics (Target Threshold = 200)

	SSA	PGA
Distinct Targets (63 in original image)	79	46
Correct Detections	57	25
False Alarms	22	21

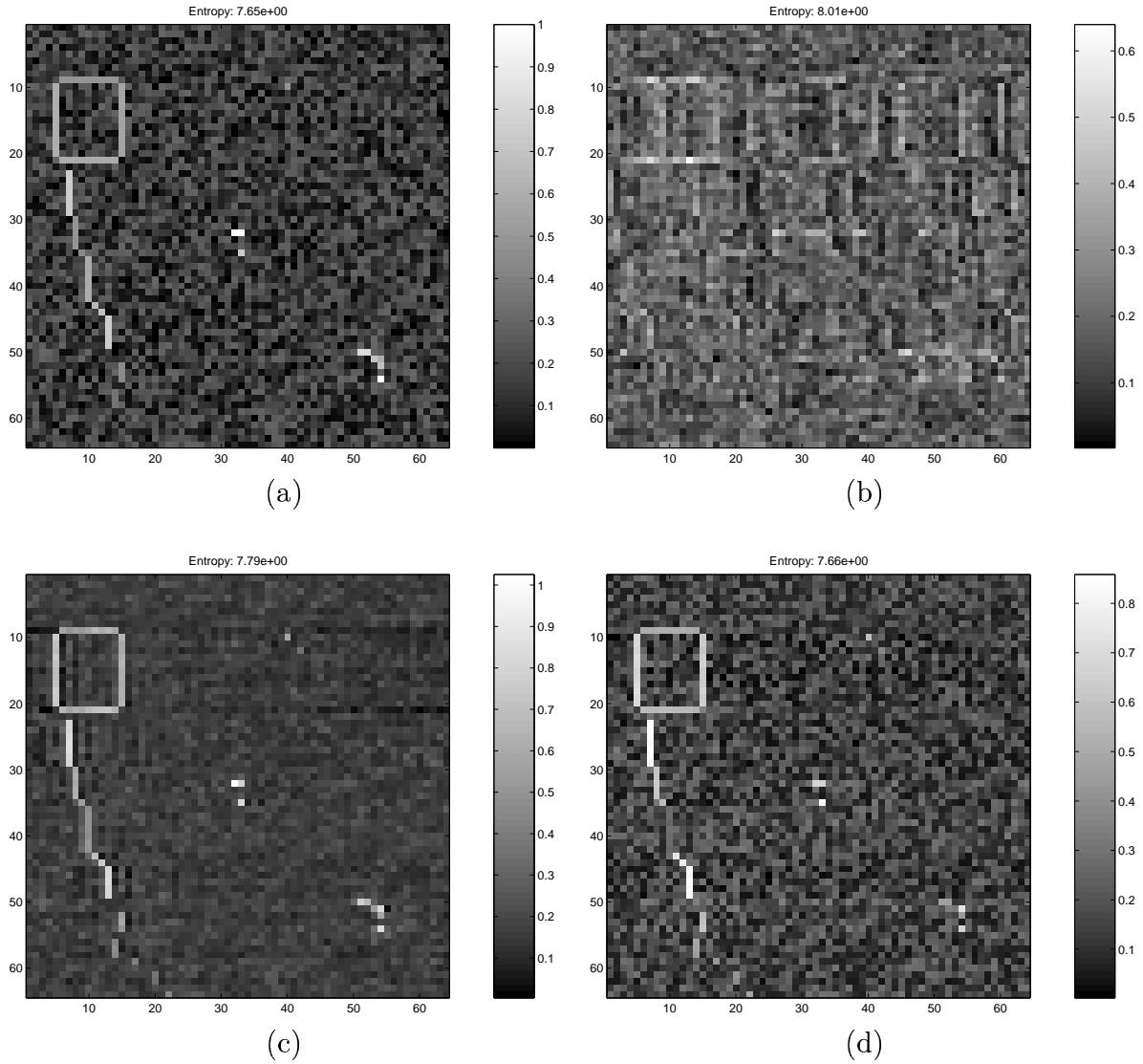


Figure 3.1 Synthesized test image (clutter range 0.30)—no random phase added in the spatial domain: (a) original image, (b) corrupted image, (c) PGA reconstruction (5 iterations), and (d) SSA reconstruction.

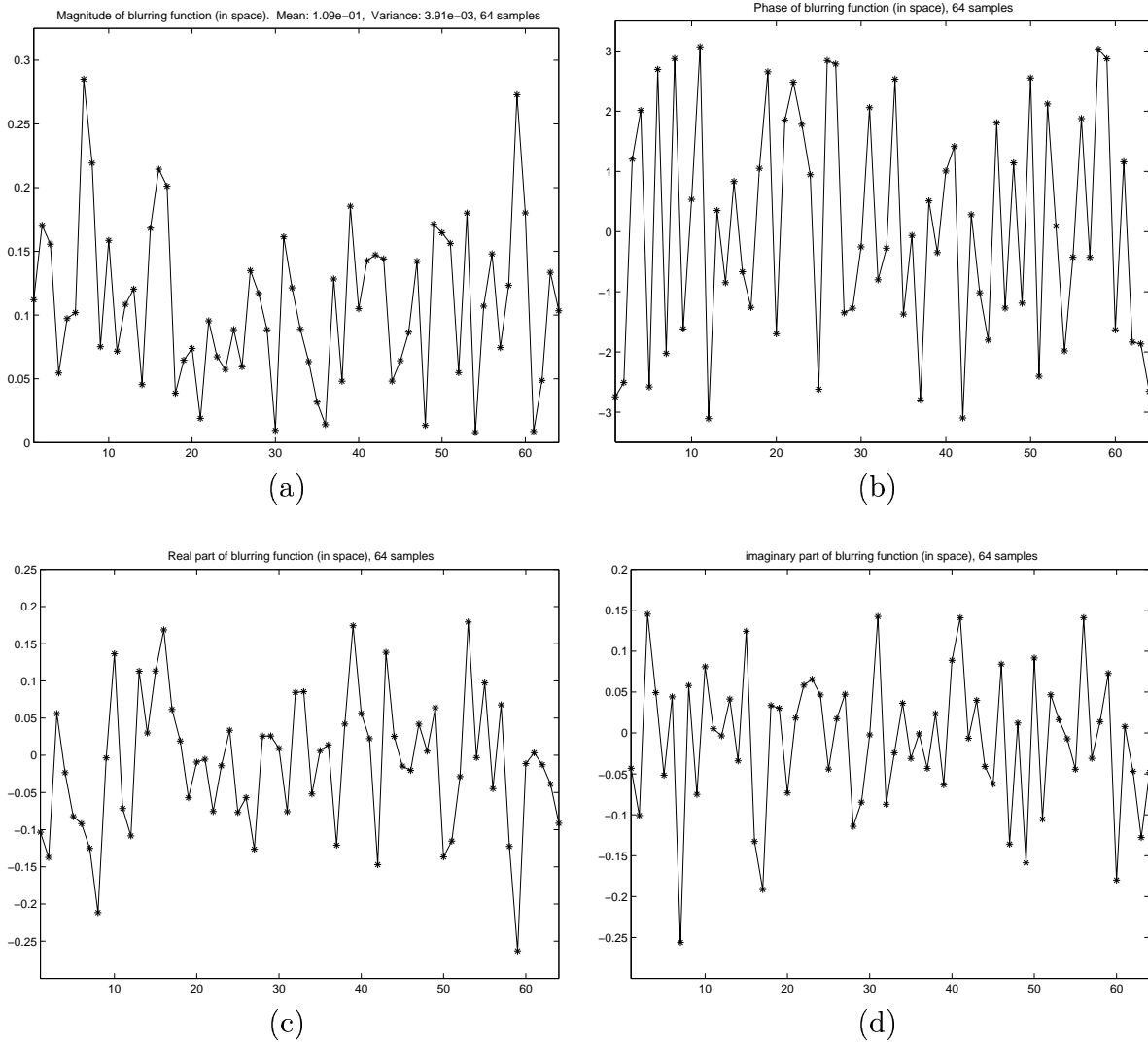


Figure 3.2 Blurring kernel arising from applying a white phase error, uniformly distributed on $-\pi$ and π , in the Fourier domain. The kernel is 64 elements in length. (a) Magnitude of the blurring kernel, Rayleigh distributed in the limit; (b) Phase of the blurring kernel, which is uniformly distributed between $-\pi$ and π in the limit; (c) Real part of the blurring kernel, Gaussian distributed in the limit; and (d) Imaginary part of the blurring kernel, Gaussian distributed in the limit.

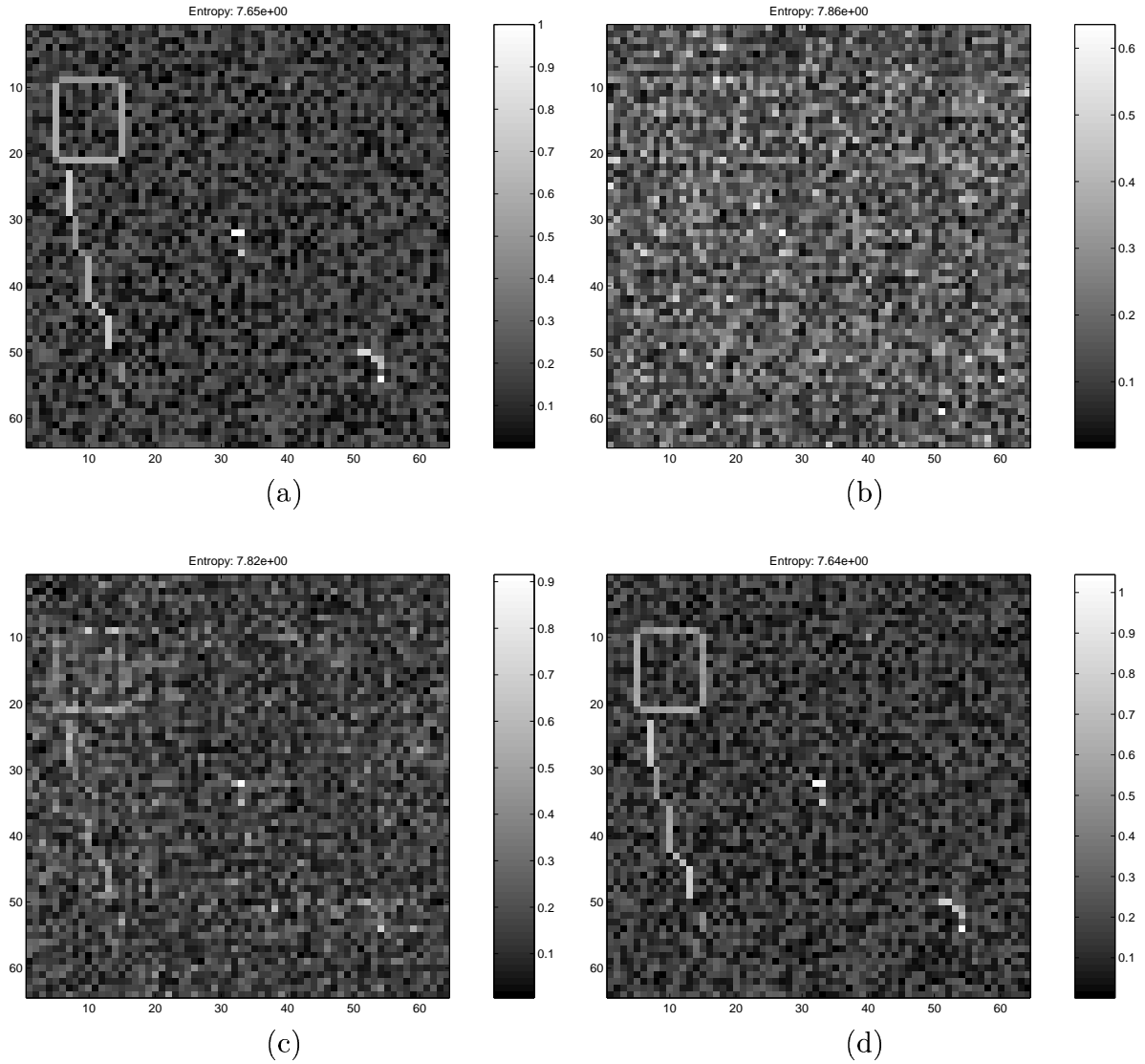


Figure 3.3 Synthesized test image (clutter range 0.30) with random phase added in the spatial domain: (a) original image, (b) corrupted image, (c) PGA reconstruction (4 iterations), and (d) SSA reconstruction.

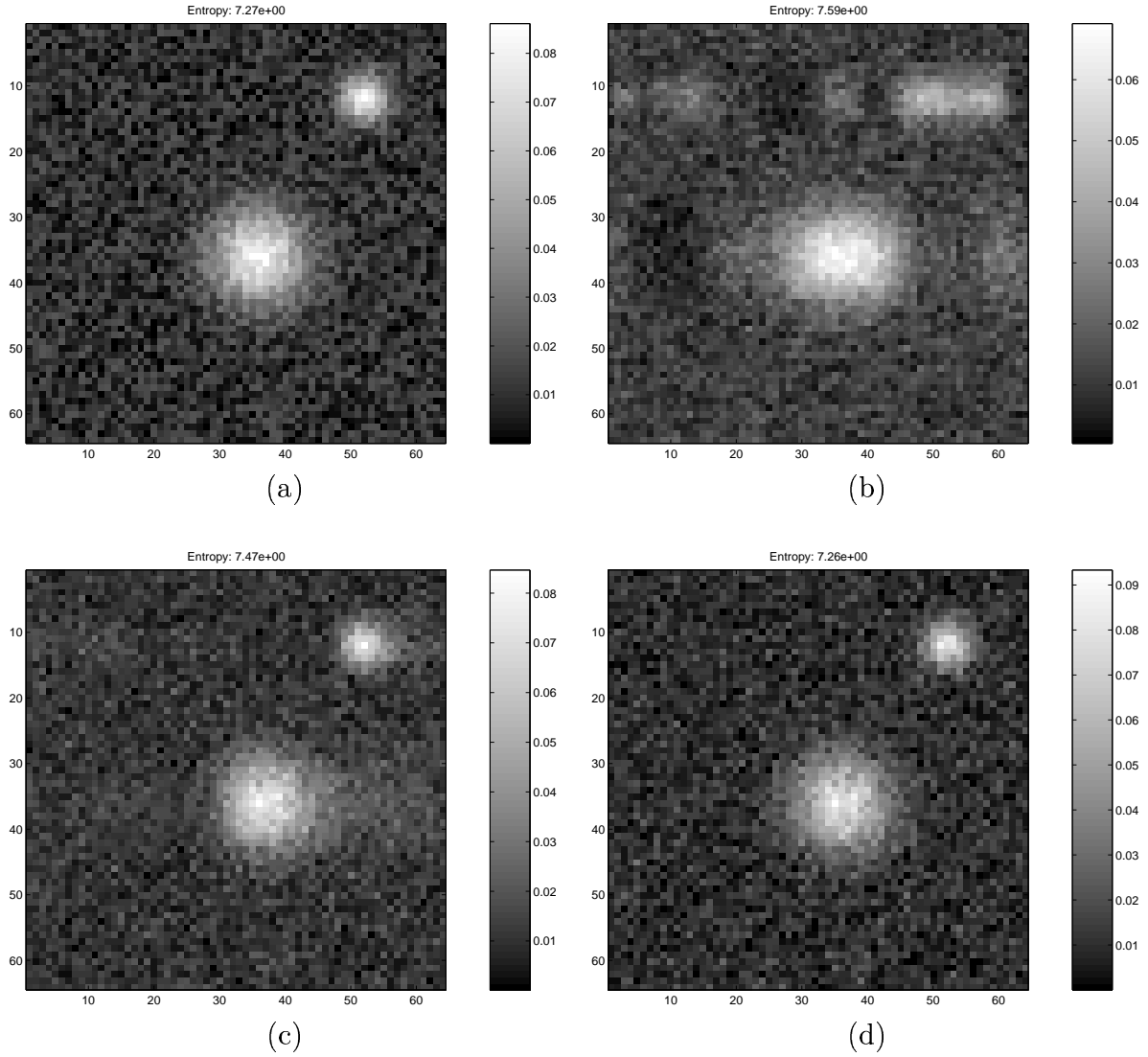


Figure 3.4 Synthesized Gaussian hill image (clutter range 0.30)—no random phase in the spatial domain: (a) original image, (b) corrupted image, (c) PGA reconstruction (2 iterations), and (d) SSA reconstruction.

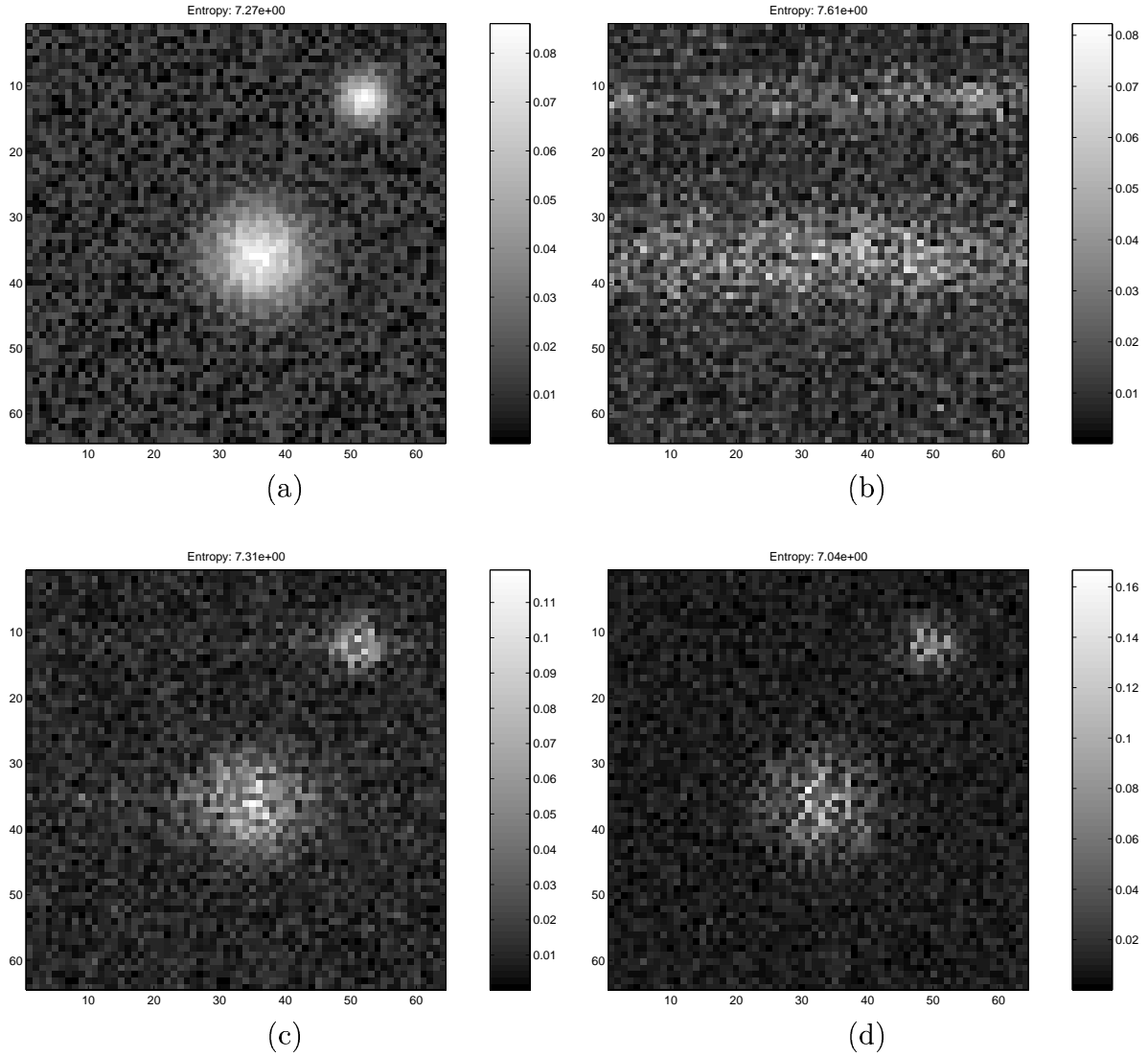


Figure 3.5 Synthesized Gaussian hill image (clutter range 0.30) with random phase in the spatial domain: (a) original image, (b) corrupted image, (c) PGA reconstruction (4 iterations), and (d) SSA reconstruction.

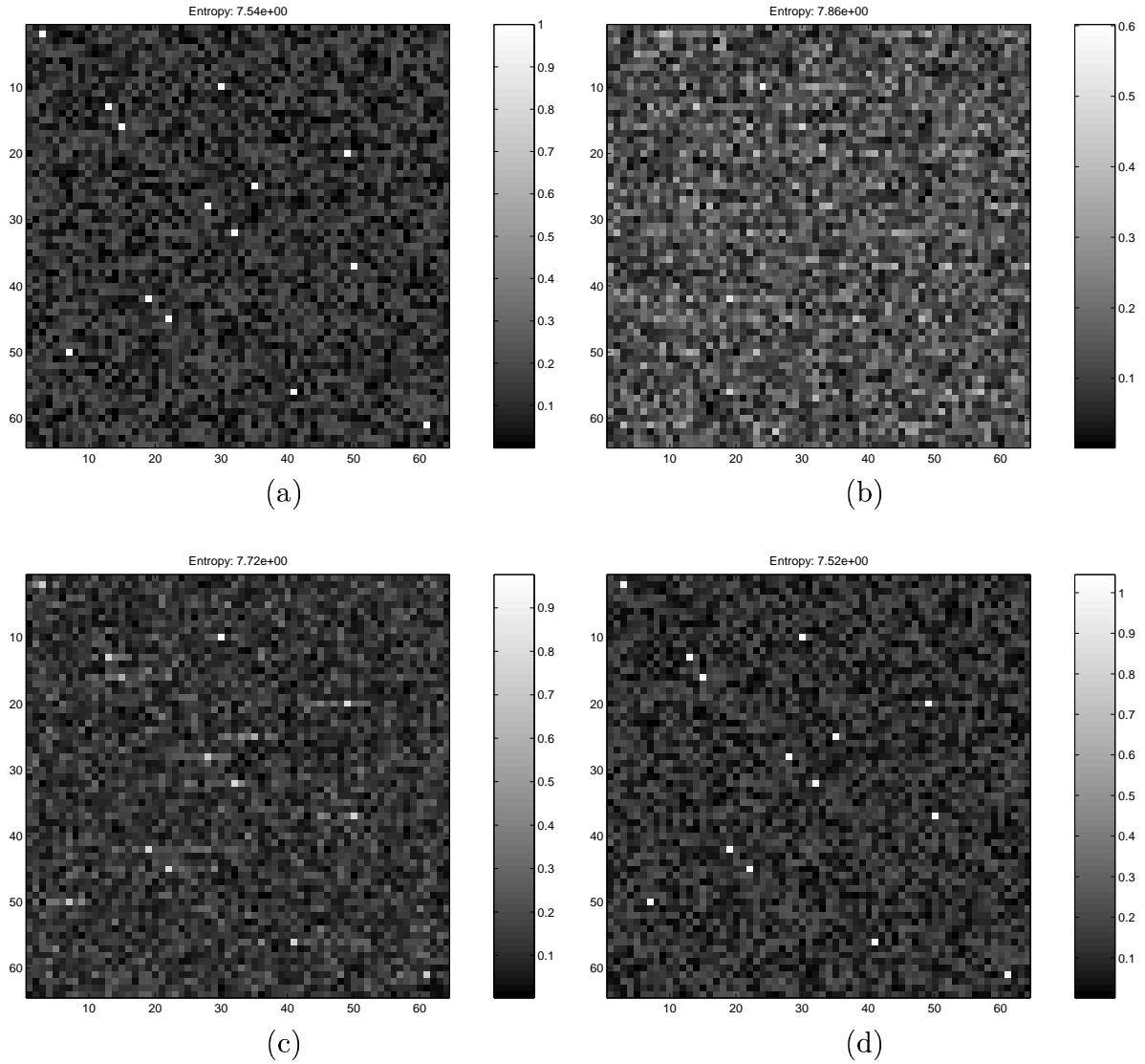


Figure 3.6 Synthesized point target image (clutter variance 0.0052) with random phase in the spatial domain: (a) original image, (b) corrupted image, (c) PGA reconstruction (7 iterations), and (d) SSA reconstruction.

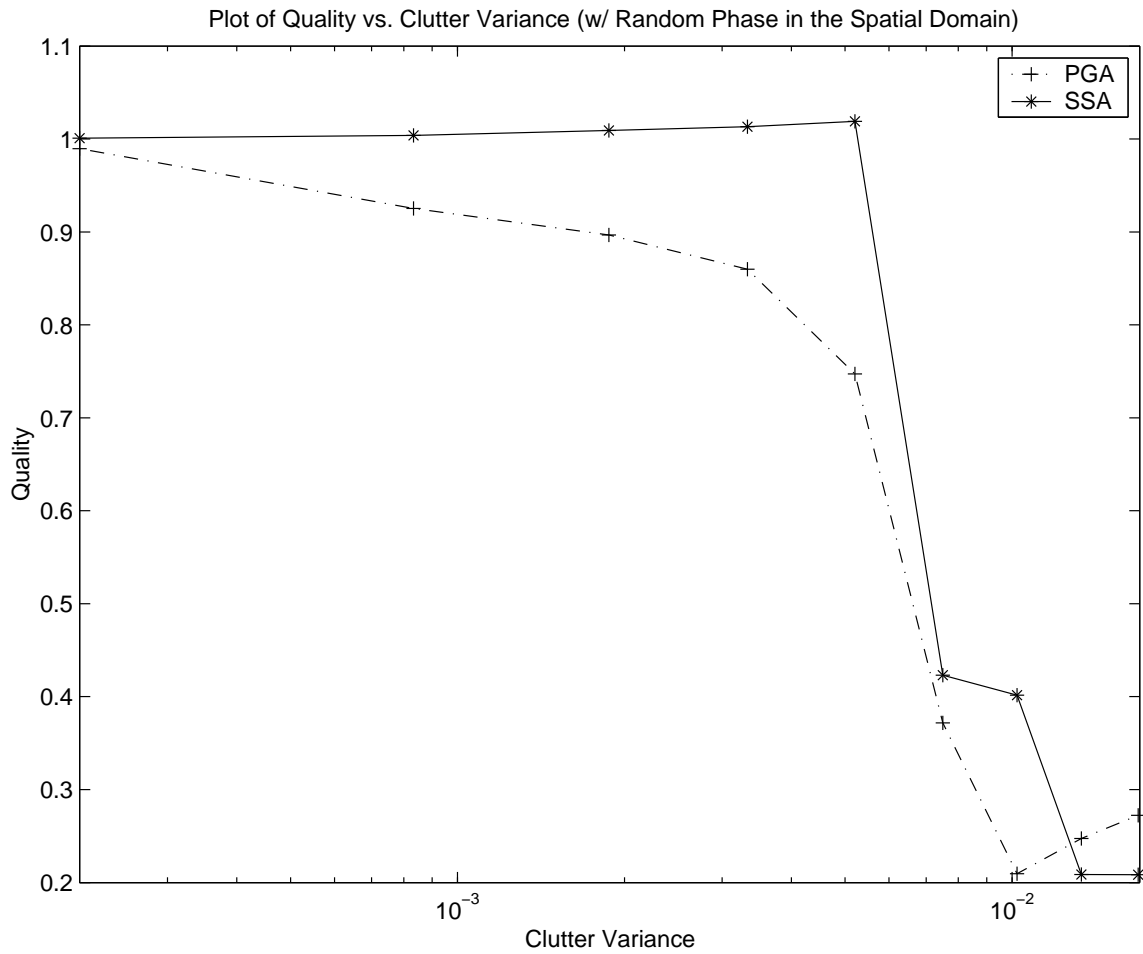


Figure 3.7 A plot of the quality measure, the average of the magnitudes of the 14 point targets in the reconstructed images of Figure 3.6, versus the input clutter variance, for both PGA and SSA.

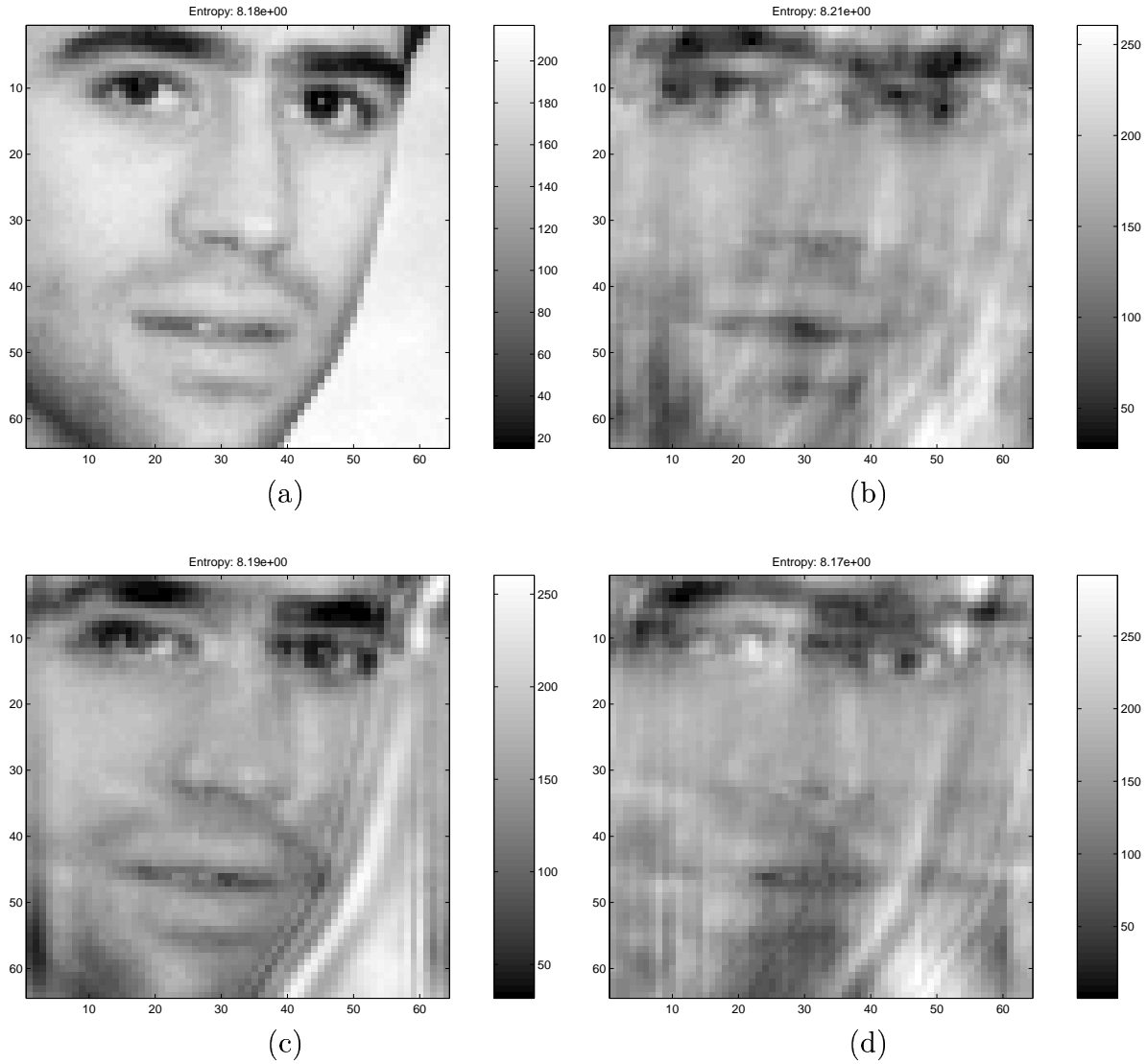


Figure 3.8 Photographic image. The corrupted image was produced by applying a white phase error, uniformly distributed on $(-\pi$ and $\pi)$, to the Fourier data of the original image: (a) original image, (b) corrupted image, (c) PGA reconstruction (7 iterations), and (d) SSA reconstruction.

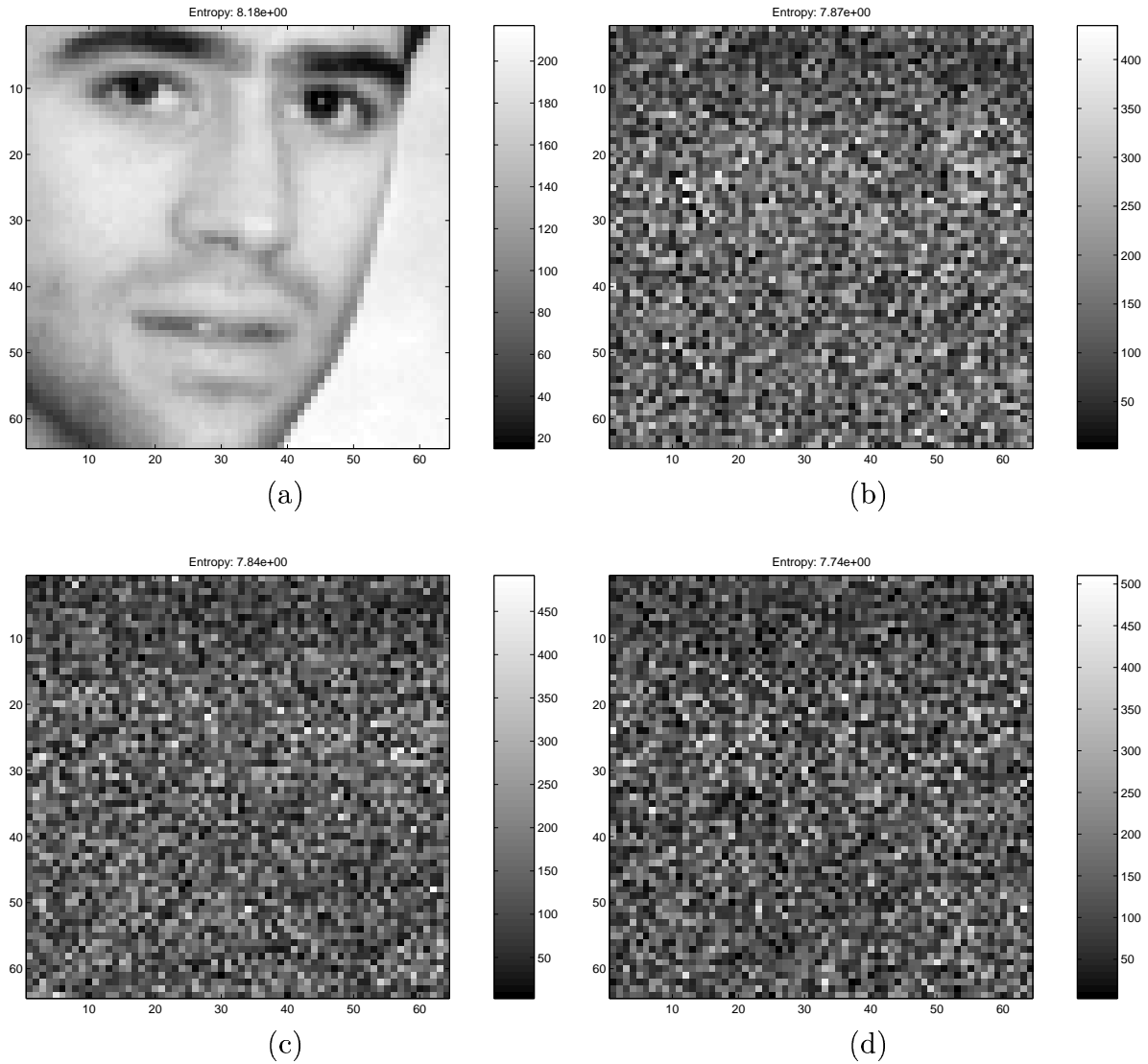


Figure 3.9 Photographic image of Figure 3.8 with random phase added in the spatial domain: (a) original image, (b) corrupted image, (c) PGA reconstruction (7 iterations), and (d) SSA reconstruction.

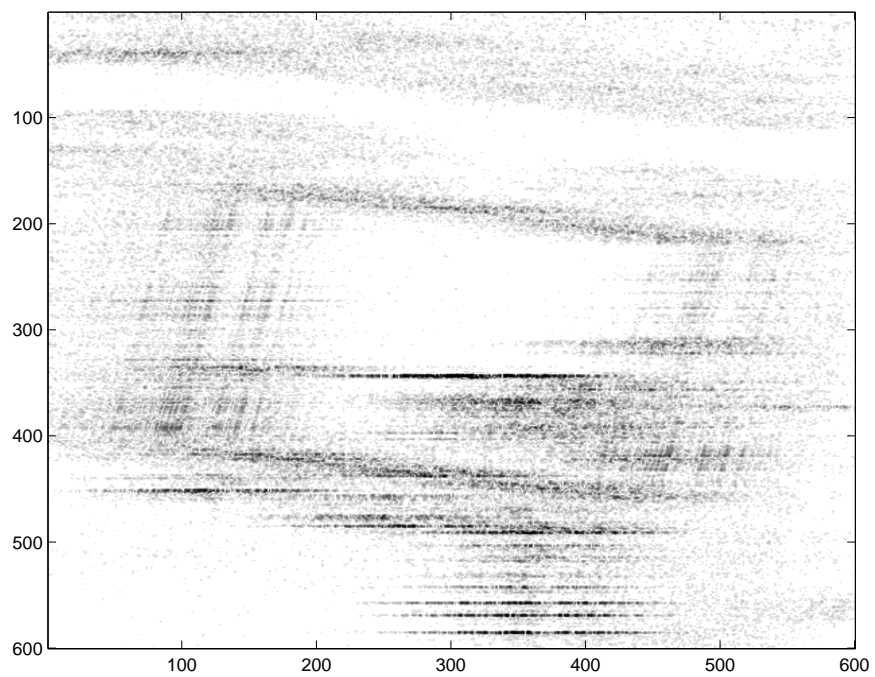


Figure 3.10 Defocused SAR image.

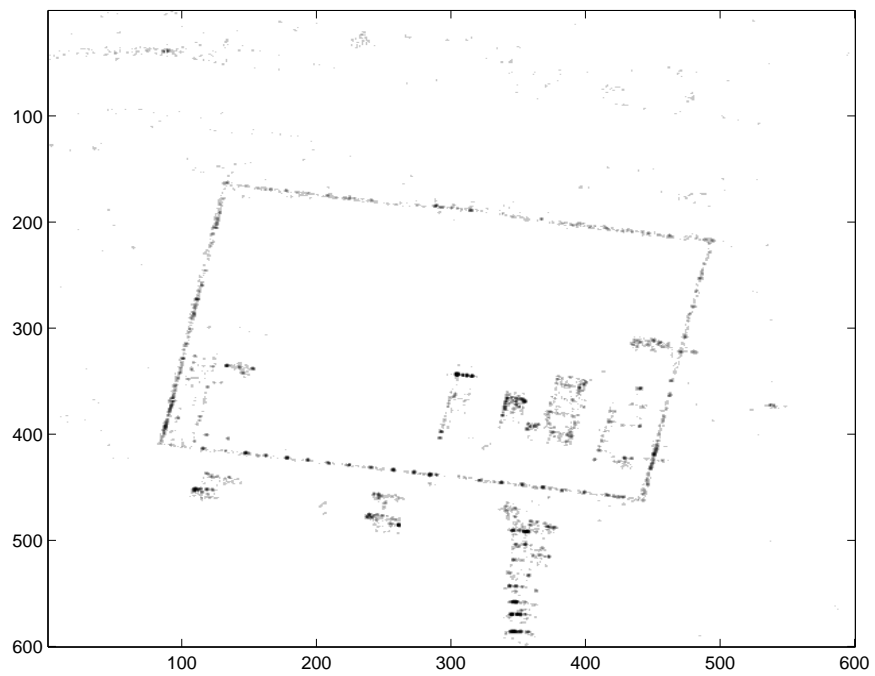


Figure 3.11 SSA reconstruction of the corrupt image in Figure 3.10.

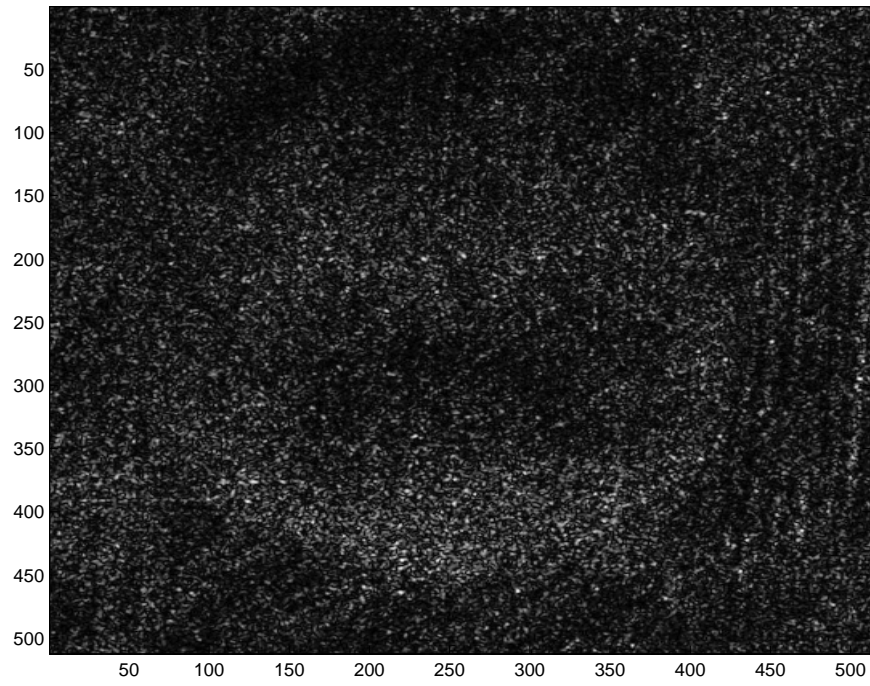


Figure 3.12 The uncorrupted SAR image of a grassy scene.

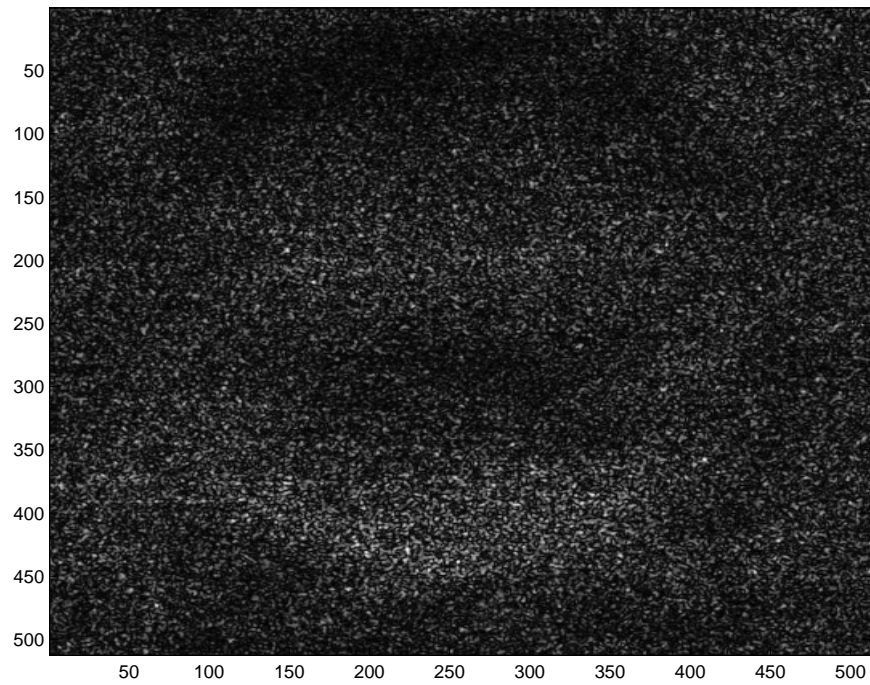


Figure 3.13 The corrupted image of the grassy scene.

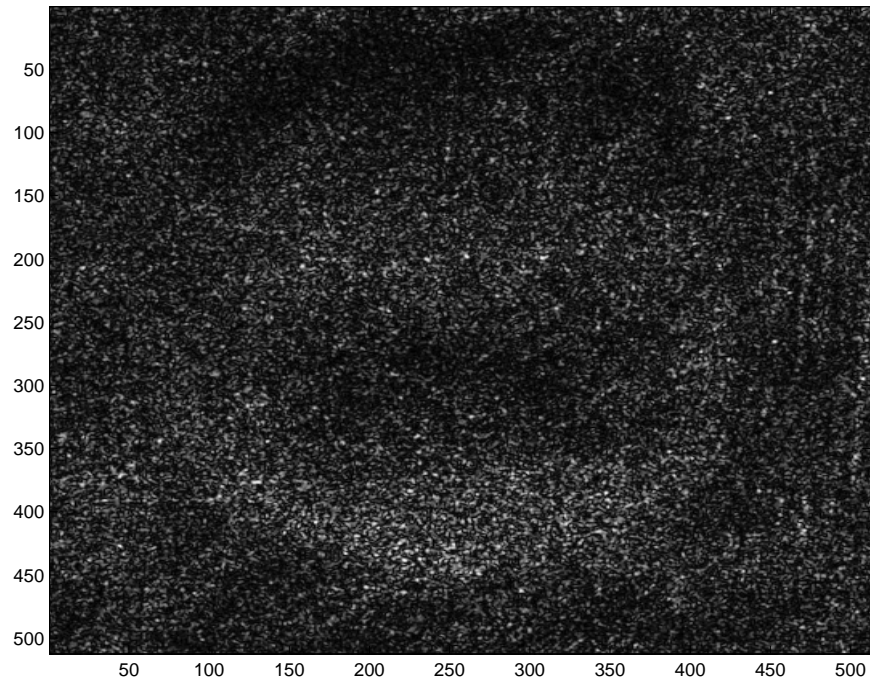


Figure 3.14 PGA reconstruction of the grassy scene.

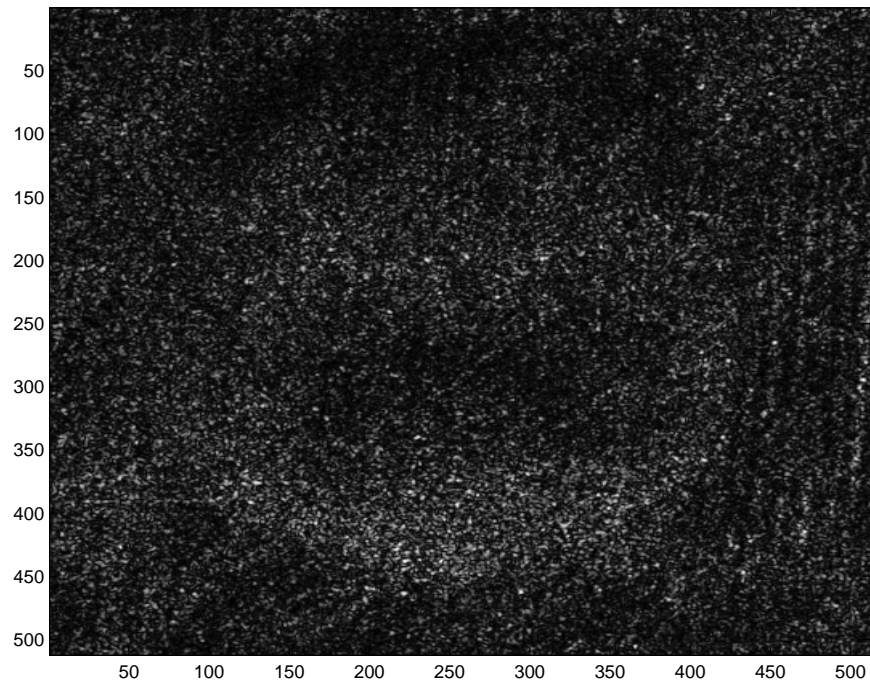


Figure 3.15 SSA reconstruction of the grassy scene.

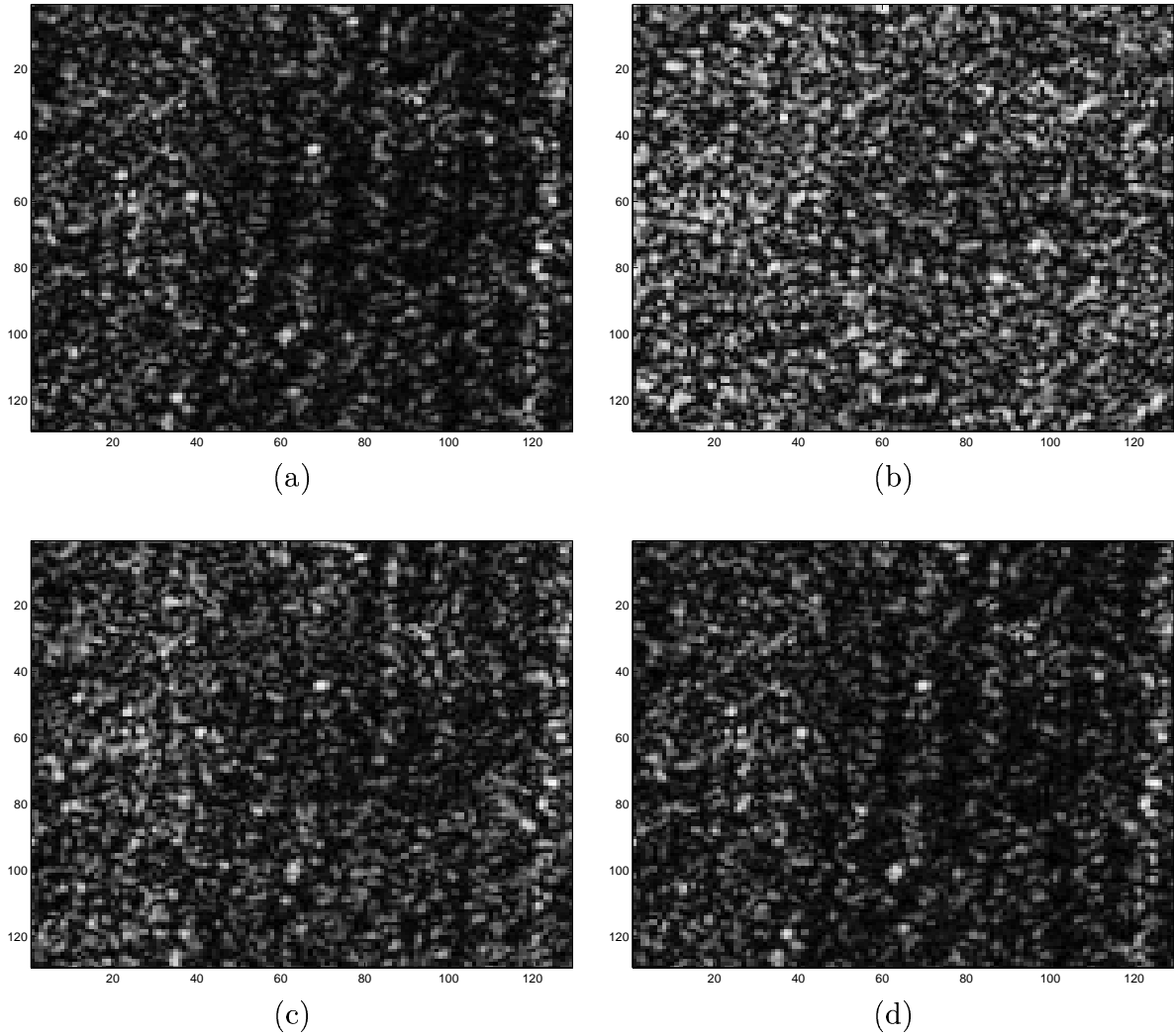


Figure 3.16 Enlarged portion of grassy SAR image (x coordinates 384 through 512, and y coordinates 222 through 350), (a) original image, (b) corrupted image, (c) PGA reconstruction, and (d) SSA reconstruction.

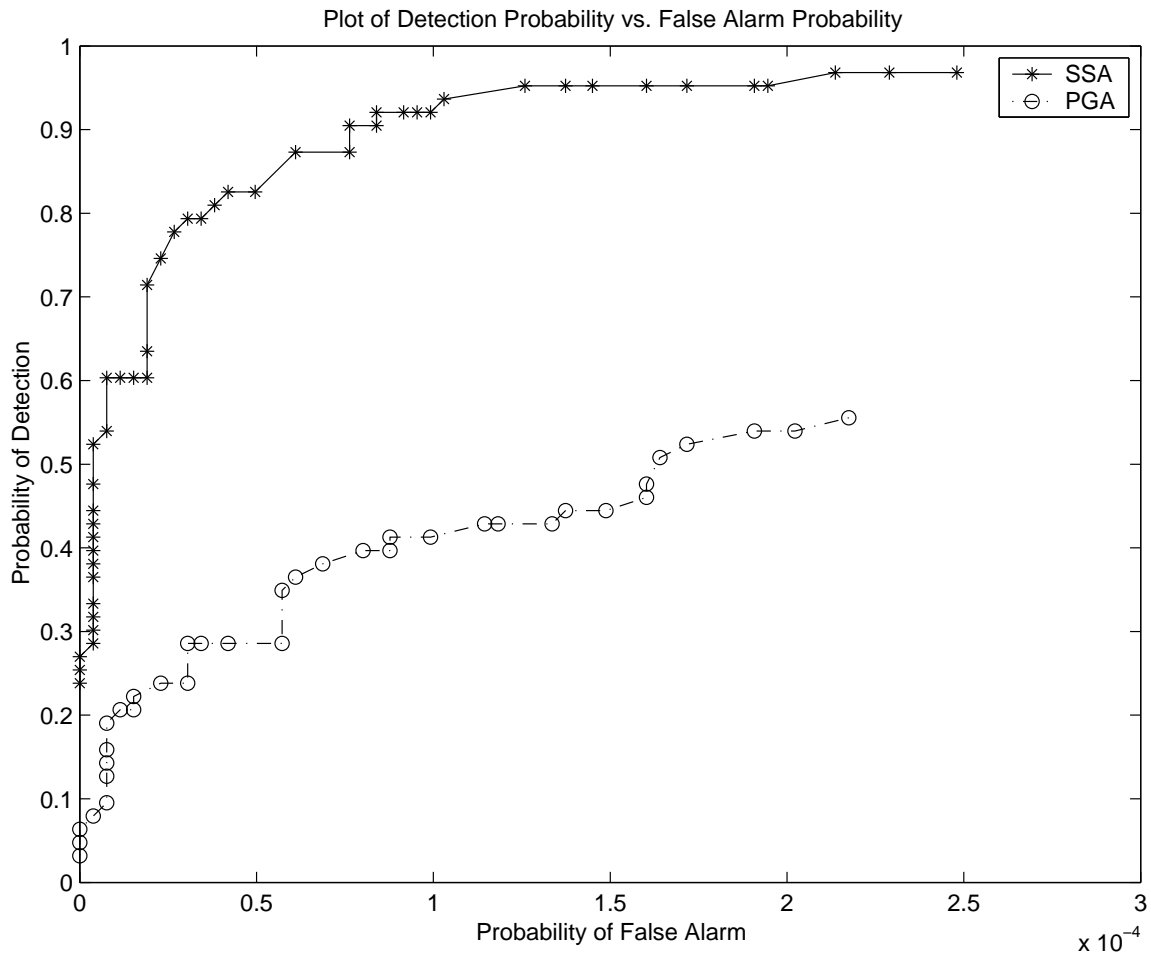


Figure 3.17 Plots of probability of detection vs probability of false alarm for targets in the grassy SAR image, for both PGA and SSA.

CHAPTER 4

EXPERIMENTS WITH SSA

The SSA approach to autofocus performs well when compared with the state-of-the-art PGA under the worst case phase error function, a random phase error. However, SSA is slow and it generally does not find an image having minimum entropy, that is, SSA does not produce an optimal result. Modifications to the search strategy can be made to speed up SSA, and to produce improved reconstructions. The problem with the search strategy is that it seeks a global minimum of the entropy function, but it may get “trapped” in a local minimum, resulting in an incorrect restoration. Modifications to SSA to improve its computational efficiency are explored in Section 4.1. The problem of finding the global minimum, and proposed solutions, are addressed in Section 4.2.

4.1 Modifications for Efficient Computation

The computational burden of SSA was considered in Section 3.3. Expediting implementation details that do not alter the SSA algorithm were presented. In this section, variations of SSA aimed at decreasing run time are explored. In this preliminary study, these variations have produced only modest time savings, and sometimes have resulted in lower quality reconstructions. The first variation considered was *multiresolution SSA*, a strategy where SSA was applied to downsampled versions of the corrupted image. Another strategy was a *hybrid PGA-SSA*, where PGA was first applied to the corrupted image to obtain a coarse estimate of the phase error. Finally, simpler cost functions,

such as the ratio of the standard deviation to the mean of the image intensity and maximum amplitude, were examined. Use of these cost functions sometimes produced poorly focused or incorrect solutions. However, in some circumstances an alternative cost produced a successful reconstruction where the entropy cost did not. These issues will be explored in the following subsections.

4.1.1 Multiresolution SSA

In multiresolution SSA, the corrupted image is downsampled by a factor D . The image must be lowpass filtered to $\frac{\pi}{D}$ before downsampling to prevent aliasing. The phase error that occupied the frequency band $(-\frac{\pi}{D}, \frac{\pi}{D})$ in the corrupted image prior to downsampling now occupies $(-\pi, \pi)$ in the downsampled image. The application of SSA to the downsampled image produces an estimate of this “stretched” phase error segment. The initial (non-downsampled) corrupted image is then lowpass filtered to $(-\frac{2\pi}{D}, \frac{2\pi}{D})$ and downsampled by a factor of $\frac{D}{2}$. The phase error from the D -downsampled image of the first stage provides an estimate of the phase error in $(-\frac{\pi}{2}, \frac{\pi}{2})$ for the $\frac{D}{2}$ -downsampled image. This process is repeated until the size of the initial corrupt image is reached.

The question of whether this method provides any computational savings depends on how the problem scales. If the amount of time SSA takes to process a downsampled image is proportional to the linear dimension of the image, this would not be a good approach. Another question is how much the computation time of later stages (i.e., the $\frac{D}{2}$ -downsampled image) is decreased by using the phase error estimate of the previous stages (i.e., the D -downsampled image). Simulations show that the multiresolution approach works for small synthesized images, and that savings are achieved. The result of running multiresolution SSA on the synthesized terrain image is presented in Figure 4.1. The 64 by 64 pixel image was initially downsampled by a factor of four. Multiresolution SSA took 13.7 s on average, while standard SSA took 43.5 s. Whether this approach works on larger, more complex images, such as the 512 by 512 pixel grassy image of Figure 3.12, remains an open question. The multiresolution approach was applied to the grassy image, but the algorithm terminated prematurely. Tweaking the termination tolerances

appropriately may circumvent this problem. This approach, when applied to images with lower signal-to-clutter ratios, was found to produce lower quality reconstructions than standard SSA. In addition, better reconstructions with faster execution times can be achieved by limiting the number of iterations or increasing tolerances T_0 and T_1 .

4.1.2 Hybrid PGA-SSA

Another expediting approach, in theory, is to use PGA to obtain a coarse estimate of the phase error function. This estimate is then input to SSA and used as the initial phase error estimate and refined. Since PGA is much faster than SSA, the motivation behind this approach is to let PGA do the meat-and-potatoes work of finding a coarse estimate of the true phase error. SSA can then be used to improve the estimate, by seeking the entropy minimum. The hope is that, given a good initial phase error estimate, SSA will quickly proceed through the first iterations (corresponding to large step sizes), and then spend most of its time searching with smaller step sizes. The SSA algorithm can be modified to initialize the step size to $\frac{\pi}{4}$ instead of π to prevent needless searching at the larger step sizes. The hybrid PGA-SSA approach was implemented, but, surprisingly, no significant decrease in run time was observed in comparison with standard SSA on the simulated terrain image of Figure 3.1. Additional study of this approach on a wider range of data must be conducted to evaluate its effectiveness.

4.1.3 Alternative cost functions

The entropy calculation after each perturbation is computationally expensive because of the natural log operator in Equation (2.19). Therefore, more time-efficient cost functions are of interest. Two costs that have been considered are the ratio of the standard deviation to the mean of the image intensity expressed in Equation (2.18), and the maximum value of the image amplitude in Equation (2.16). The first cost function will be referred to as the *intensity-squared cost*, since it comprises a sum of the squared intensities, and the second cost will be denoted the *maximum amplitude cost*.

These costs were incorporated into SSA, and each resulting variant of the algorithm was applied to the terrain image. The terrain image had a clutter range of 0.15 and random phase applied in the spatial domain. Use of the entropy function produced a reconstruction in an average of 36.2 s, while the intensity-squared cost took 23.6 s and the maximum amplitude cost required 50.8 s. The intensity-squared cost required fewer cycles, because it is essentially the same as the entropy calculation, but without the natural log. The maximum amplitude required more, since each pixel of the image had to be searched to determine the absolute maximum. The MATLAB “max” function was used to determine the maximum pixel value. It is possible that the implementation of this function could be made more efficient. The image resulting from the intensity-squared cost was of similar quality to that produced using the entropy cost, with only minor degradation in the former. The maximum amplitude cost produced an incorrect reconstruction of the terrain image.

The alternative cost functions were found to be an appropriate for certain images, and inappropriate for others. Figure 4.2(a) depicts the terrain image with a clutter range of 0.25 and random phase applied in the spatial domain. Reconstructions using the entropy, intensity-squared, and maximum amplitude costs are shown in Figures 4.2(c) through (e), respectively. Entropy is seen to be an appropriate cost for this image. The intensity-squared and maximum amplitude costs produced poor reconstructions, where only “ghosts” of the targets are evident.

Figure 4.3(a) shows the point target image with a clutter range of 0.25 and random phase applied in the spatial domain. The reconstructed images produced using the three cost functions are presented in Figures 4.3(c) through (e). Here, the entropy and intensity-squared costs both produced successful reconstructions. The maximum amplitude cost produced a lower quality reconstruction, but the point targets are all still evident.

A harder test case to reconstruct was the point target image with a clutter range of 0.35 and random phase applied. This image is displayed in Figure 4.4(a). Standard SSA using the entropy cost function could not reconstruct this image, as Figure 4.4(c) shows.

Surprisingly, SSA with the intensity-squared cost did recover this image, as depicted in Figure 4.4(d). Recent work by Fienup and Miller [10] supports this finding. The effectiveness of a cost function on a particular image can be determined by analyzing the second derivative of the cost with respect to the image intensity. The intensity-squared cost is predicted to work best on images having bright isolated point scatterers, while entropy is favored for images lacking bright isolated scatterers, but containing tree-like features and shadowy areas.

The variant of SSA incorporating the intensity-squared cost was applied to the grassy image of Figure 3.13. An incorrect reconstruction, consisting of one point target against a constant background, resulted. Fienup and Miller [10] have reported similar findings, stating “Use of larger powers [such as the intensity-squared metric] for [rural images lacking bright isolated scatterers] can result in stagnation at a false maximum sharpness that concentrates too much energy at a single point in the image.”

It would be of interest to consider other cost functions that can be accurately approximated by functions that are easily updated. Since only one element of the phase error estimate is perturbed at a time, it would be efficient to compute a cost function increment between perturbations, as opposed to having to compute the full cost. Since the entropy function is nonlinear due to the log term, it is not a good candidate for such an approach. However, approximating the log by a piecewise linear function could work. Further investigation of this topic would be worthwhile.

4.2 Variations of the SSA Search Strategy

The SSA algorithm searches for the phase error estimate $\hat{\phi}$ that minimizes an M -dimensional entropy function, where M is the number of elements in $\hat{\phi}$. However, this entropy function is known to have local minima. A successful algorithm is one that determines the global minimum of entropy. This problem is most easily visualized by considering a 2-D surface with several valleys and hills. One valley is the deepest, and the problem becomes that of how to search the surface in a systematic way to find the deepest

valley. Standard SSA always takes steps in the direction of the lowest entropy, in one dimension at a time. Thus, once the algorithm gets stuck in a local minimum, it has no way of getting out. In practice SSA can become trapped in local minima, especially when images with low signal-to-clutter ratios are considered. As a consequence, we investigate variations on the SSA searching strategy.

The first class of alternative searching algorithms are deterministic. These algorithms produce a unique output for a particular input and do not rely upon a random factor. Three deterministic algorithms were considered: a multidimensional approach, where multiple phase dimensions are perturbed simultaneously; a nonmonotonic step size evolution approach (called *variable step size SSA*), where the step size alternately decreases and increases with each iteration according to a particular schedule; and an approach where phase samples are perturbed and then revisited several times before a subiteration commences, which is denoted *perturbation round SSA*. Of these three, variable step size SSA is the only algorithm that can escape from local minima. The other two provide a more extensive search at the expense of more computation time.

The second class of searching algorithms is *probabilistic*. These algorithms use random parameters. For a given input, the output of the algorithm is not unique. The problem with most deterministic search strategies is that they lack mechanisms for escaping local minima. The probabilistic algorithms are based on a procedure called *simulated annealing*, where the search algorithm is allowed to take steps in the “wrong” direction, namely uphill steps, with a specified small probability that decreases with each iteration. Thus, the algorithm has a chance to escape local minima. The deterministic and probabilistic search strategies are described in detail in the following subsections.

Before investigating alternative search techniques, it is of interest to determine how well a corrupted image can be restored if the step size is limited to a particular value. For example, what is the best image that can be produced if only steps of size π are taken? A conclusive answer has not been found to date, but a good way of determining how well an image can be fixed with a particular step size is to quantize the true phase error to that step size and then apply the complex conjugate to the corrupted image. In

Figure 4.5(c), the phase error that produced the corrupted image of Figure 4.5(b) has been quantized to step size π , conjugated, and applied as the phase error estimate. This image reveals some of the point targets upon careful inspection. If the true phase error is quantized to $\frac{\pi}{2}$, conjugated, and applied to the corrupted image, the image in Figure 4.5(d) results. This image compares well to the original image. Thus it seems possible to obtain a decent reconstruction if only steps of size $\frac{\pi}{2}$ are taken. Figure 4.5(e) shows that SSA did not converge to the true phase error.

There is a problem with the minimum entropy approach. For images with high signal-to-clutter ratios, the minimum entropy image will correspond to the focused image because the best way to perturb the phases such that the lowest entropy image is achieved is to place the image in its original configuration. In other words, when the underlying image has a lot of structure, it is unlikely that the phases can be perturbed in such a way that a sharper false image is formed. However, when the signal-to-clutter ratio is low, there are may be several ways the phases can be perturbed to produce images with entropies equal to or lower than the original image. Therefore, the lowest entropy image may correspond to a completely erroneous reconstruction. Figure 4.6 shows an instance where the original image and a completely erroneous SSA reconstruction have similar costs, and an image corrected with the true phase error quantized to $\frac{\pi}{2}$ that visually resembles the original image has a higher cost. In spite of this problem, SSA produces high-quality reconstructions in a wide variety of imaging scenarios, and the robustness of the algorithm increases when variations to the standard searching strategy are implemented.

4.2.1 Deterministic algorithms

4.2.1.1 Multidimensional SSA

In the multidimensional SSA scheme, N phase samples are perturbed simultaneously. For example, if $N = 2$, then there are nine possible moves in entropy space. In general,

there are 3^N possible moves. This search is beneficial because the algorithm moves in N dimensions at a time, which could offer better choices for minimizing the entropy.

Multidimensional SSA has been implemented for $N = 2$. Two samples were perturbed at a time, with no overlap. The result of applying Multidimensional SSA to the point image is shown in Figure 4.7(d). Many of the point targets in the original image were successfully reconstructed. Figure 4.7(c) reveals that standard SSA was unable to reconstruct the image.

4.2.1.2 Variable step size SSA

In variable step size SSA, the initial step size is set to π in the first iteration, and is then reduced to $\frac{\pi}{2}$ and $\frac{\pi}{4}$ with the second and third iterations, respectively. However, instead of the step size again decreasing by a factor of two on the fourth iteration as in standard SSA, it is reassigned to $\frac{\pi}{2}$. This pattern repeats, where the sequence of step sizes for first nine iterations is $\{\pi, \frac{\pi}{2}, \frac{\pi}{4}, \frac{\pi}{2}, \frac{\pi}{4}, \frac{\pi}{8}, \frac{\pi}{4}, \frac{\pi}{8}, \frac{\pi}{16}\}$. The benefit of revisiting larger step sizes is that SSA has the possibility of stepping out of local minima.

Figure 4.7(e) shows the (incorrect) reconstruction produced by variable step size SSA. In practice, this approach usually bypasses iterations where the step size increases; no entropy difference is detected, and the algorithm proceeds to the next iteration.

4.2.1.3 Perturbation round SSA

In perturbation round SSA, the first eight samples are perturbed in sequence. Then, instead of perturbing the ninth sample, the first eight samples are perturbed again, along with the next eight samples. This pattern repeats, with eight new samples added in each “perturbation round.” The best phase error estimate is first found among a few dimensions, and then more new dimensions are gradually included while previous ones are continually revisited. This process continues until all the phase elements have been considered. The benefit of this method is that the algorithm locks on to a particular region where the global minimum is believed to be, which makes searching in the latter dimensions more effective.

The perturbation round strategy works extremely well in practice. Figure 4.7(f) shows the image reconstruction produced using this strategy. This image is virtually indistinguishable from the original image. Though this result is promising, it does not imply that the perturbation round approach will work for all similarly generated white phase error functions. Some randomly generated errors cause the original image to be more defocused, and thus the reconstruction problem becomes more difficult.

4.2.2 Simulated annealing algorithm

None of the strategies described above provides a direct means for escaping local minima. We now think about approaches that intentionally *maximize* the entropy after a perturbation, according to the value of a randomly generated parameter. These approaches are based on the *simulated annealing* optimization procedure commonly utilized in situations where it is desired to minimize complicated multivariate functions [11].

Natural annealing is the process by which substances are heated, and then slowly cooled, to obtain a strong crystalline structure. Simulated annealing is analogous to this physical process, where the *ground state* at which the crystalline structure is obtained corresponds to the minimum of a particular cost function. In a natural annealing process, the temperature is set to a high initial value. It is then decreased slowly. At a high temperature, the molecules of a substance have a large kinetic energy and can visit many different energy configurations. If the temperature is quickly decreased, the molecules suddenly lose their kinetic energy, and the state gets trapped in an energy well that may not correspond to the global minimum. However, if the temperature is slowly decreased, the state has the opportunity to visit many energy configurations. As the temperature progressively decreases, the algorithm converges to a particular energy neighborhood.

An annealing approach can be applied to SSA. The temperature is analogous to the step size; a large step size is initially selected that allows the algorithm to visit a variety of states. The step size, like the temperature, is progressively decreased with each iteration. The minimum entropy sought by the SSA algorithm is analogous to the energy minimum

that annealing seeks. To simulate the annealing process within the SSA framework, we have devised an algorithm that we refer to as *Modified Simulated Annealing*.

We will now discuss the pertinent mechanisms of the Modified Simulated Annealing algorithm. The entropies resulting from the three candidate phase perturbations are computed after each image update. If the minimum entropy is achieved by *not* perturbing the phase element, a positive or negative step may be taken with equal probability. In our implementation, this is accomplished by constructing a random variable uniformly distributed between -1 and 1, and taking a step if the absolute value of this variable falls below an *annealing threshold*. The annealing threshold is a parameter that is initialized to a particular value, and then decreased as the algorithm progresses. If the random variable is positive, the phase element is incremented. If the random variable is negative, the phase element is decremented. The phase element is not perturbed if the random variable is greater than the annealing threshold.

If either a positive or negative step minimizes the entropy, a step in the *opposite* direction is taken if a random variable uniformly distributed between zero and one falls below the annealing threshold. If the random variable is greater than the threshold, a step is taken in the direction that minimizes the entropy.

An added innovation of our annealing approach lies in the *annealing schedule*, or the manner in which the annealing threshold is varied. Instead of starting with a high annealing threshold and then decreasing it monotonically with each subiteration, our approach resets the annealing threshold to a variable base value at the beginning of each iteration. This base value decreases with each iteration by $2^{\beta-1}$, where β represents the iteration number. Figure 4.8(d) shows the reconstruction produced using Modified Simulated Annealing. Standard SSA was unable to produce a viable reconstruction in this case.

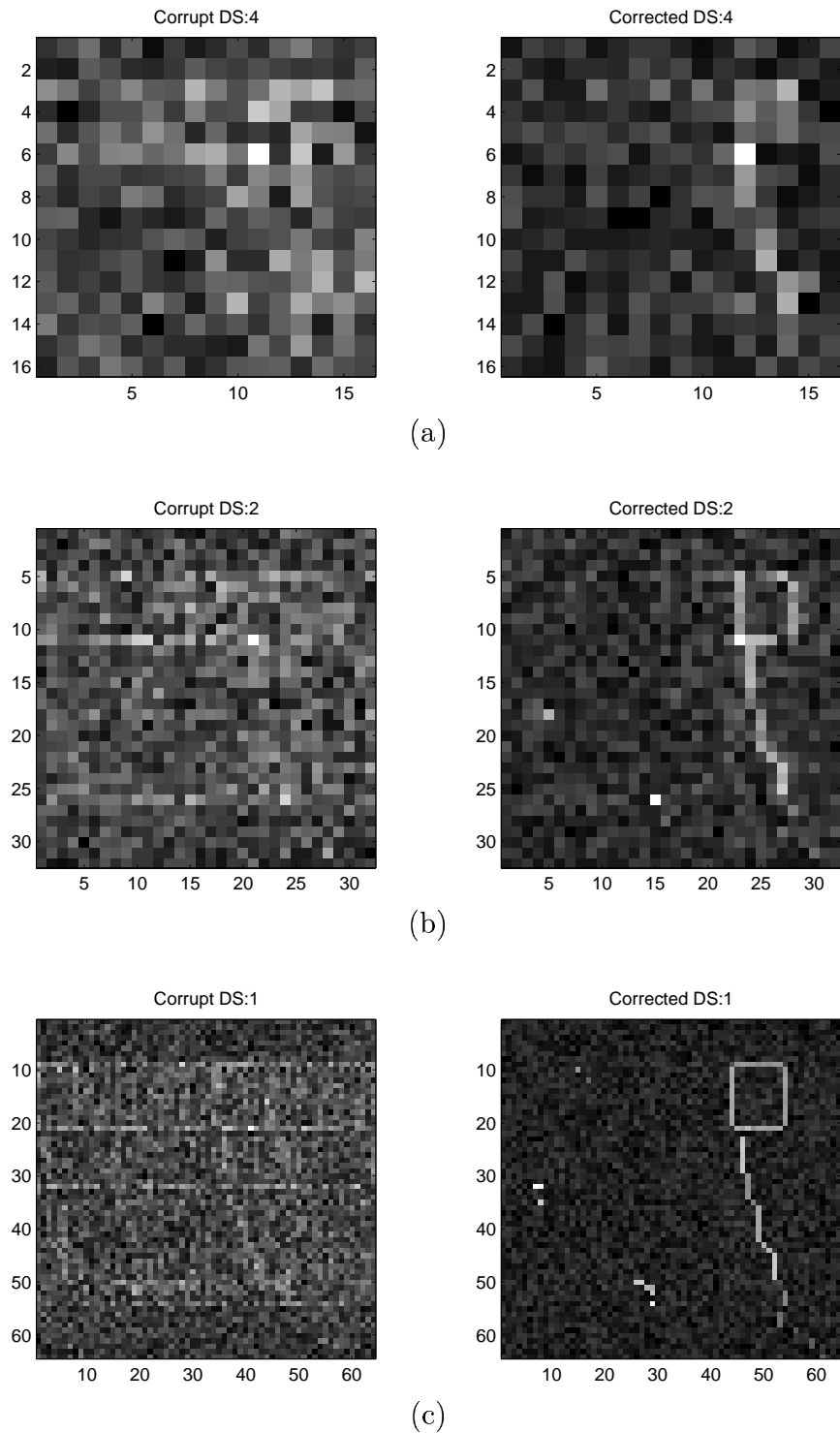


Figure 4.1 Multiresolution SSA (clutter range of 0.15, random phase in the spatial domain, initial downsampling factor of 4): (a) downsampled by 4, (b) downsampled by 2, and (c) initial corrupt image (left) and final result of multiresolution SSA (right).

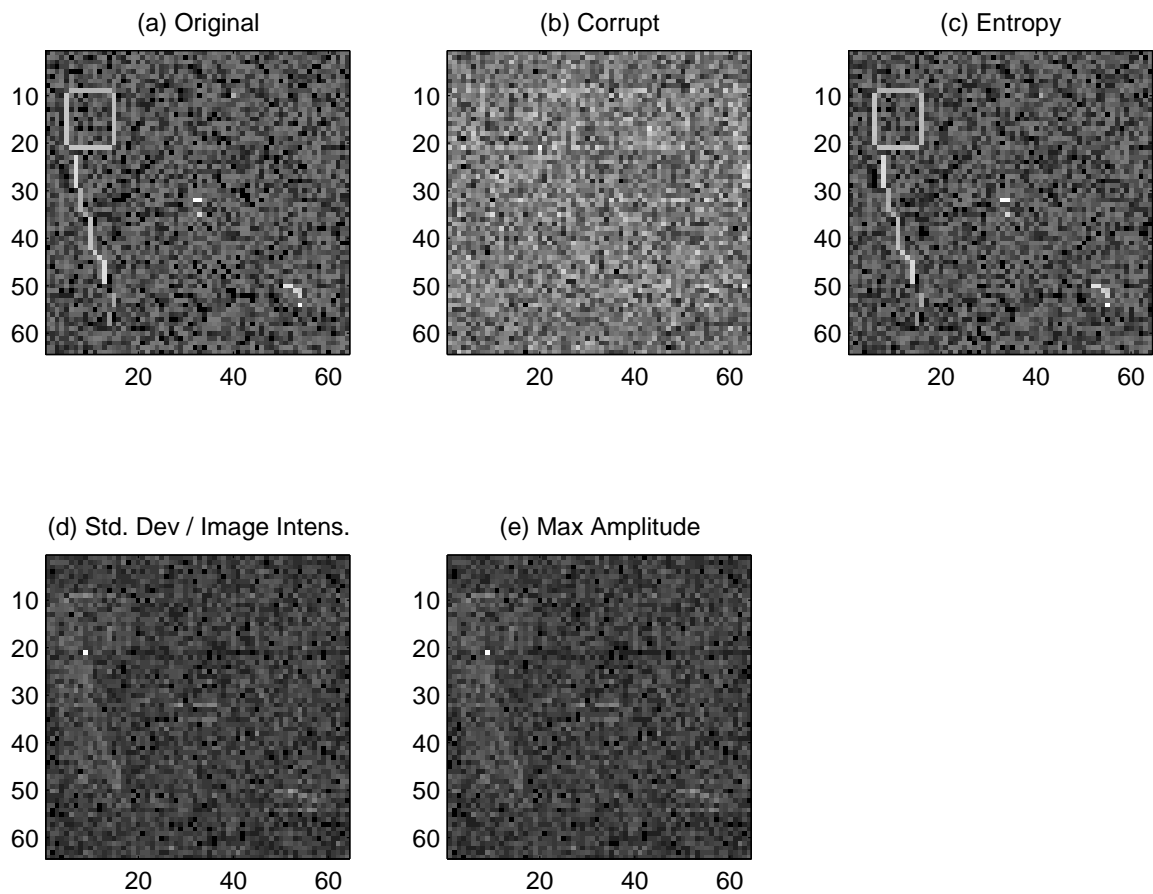


Figure 4.2 Comparison of cost functions on a synthesized terrain image, with a clutter range of 0.25 and random phase applied in the spatial domain: (a) original image, (b) corrupt image, (c) SSA with entropy cost, (d) SSA with intensity-squared cost, and (e) SSA with maximum amplitude cost.

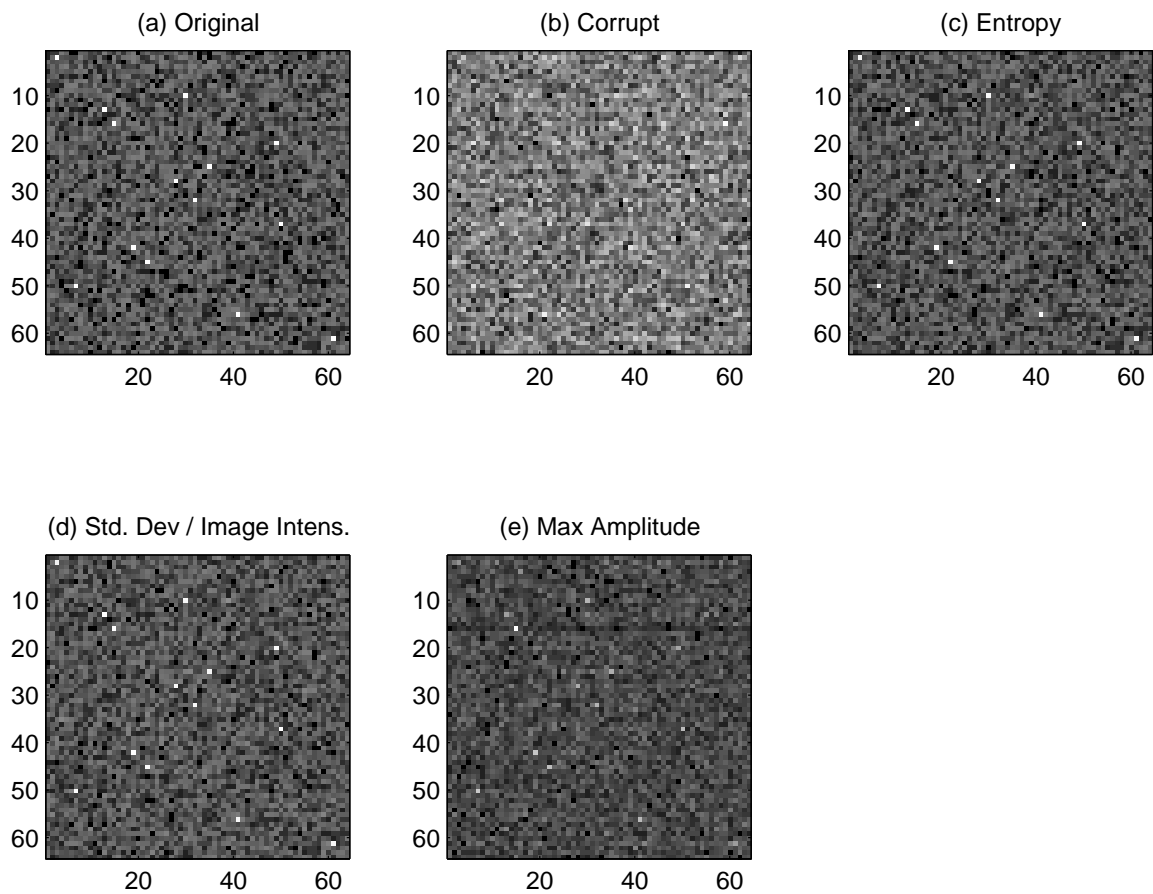


Figure 4.3 Comparison of cost functions on a synthesized point-target image, with a clutter range of 0.25 and random phase in the spatial domain: (a) original image, (b) corrupt image, (c) SSA with entropy cost, (d) SSA with intensity-squared cost, and (e) SSA with maximum amplitude cost.

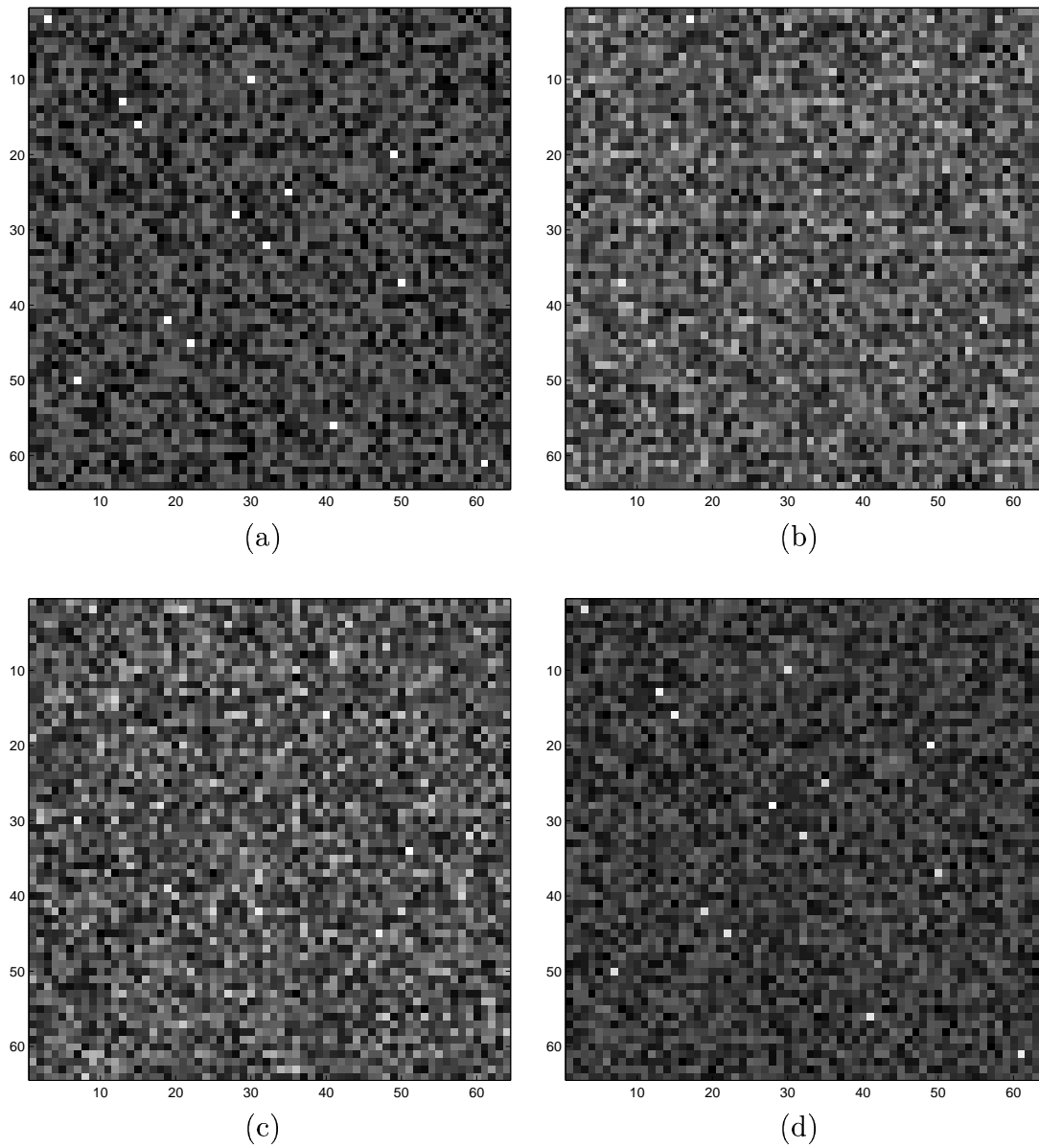


Figure 4.4 Synthesized point-target image, with a clutter range of 0.35 and random phase in the spatial domain: (a) original image, (b) corrupted image, (c) entropy cost, and (d) intensity-squared cost.

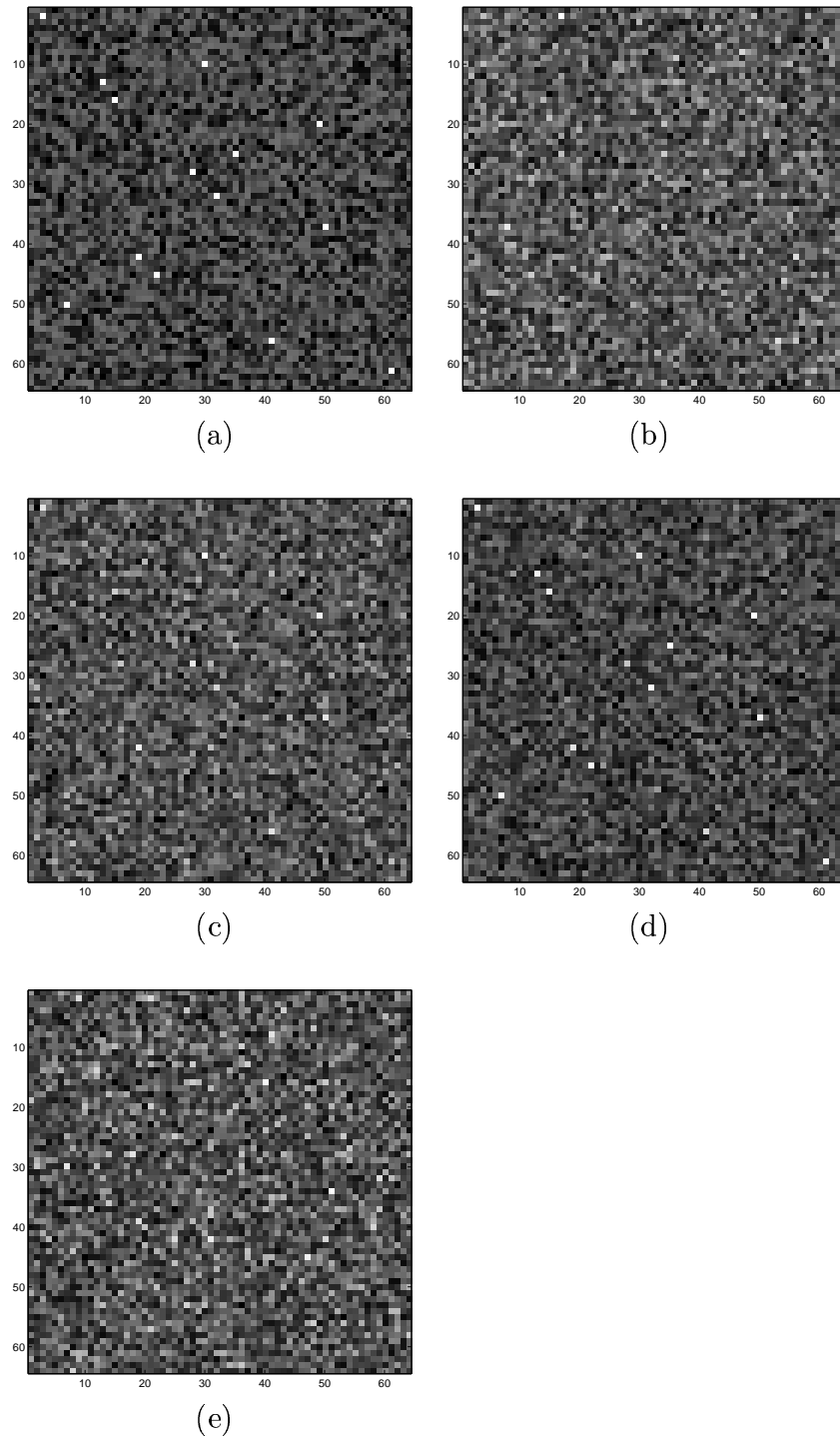


Figure 4.5 Synthesized test image with a clutter range 0.35 and random phase in the spatial domain: (a) original image, (b) corrupted image, (c) phase error estimate is the complex conjugate of the true phase error quantized to π , (d) phase error estimate is the complex conjugate of the true phase error quantized to $\frac{\pi}{2}$, and (e) standard SSA reconstruction.

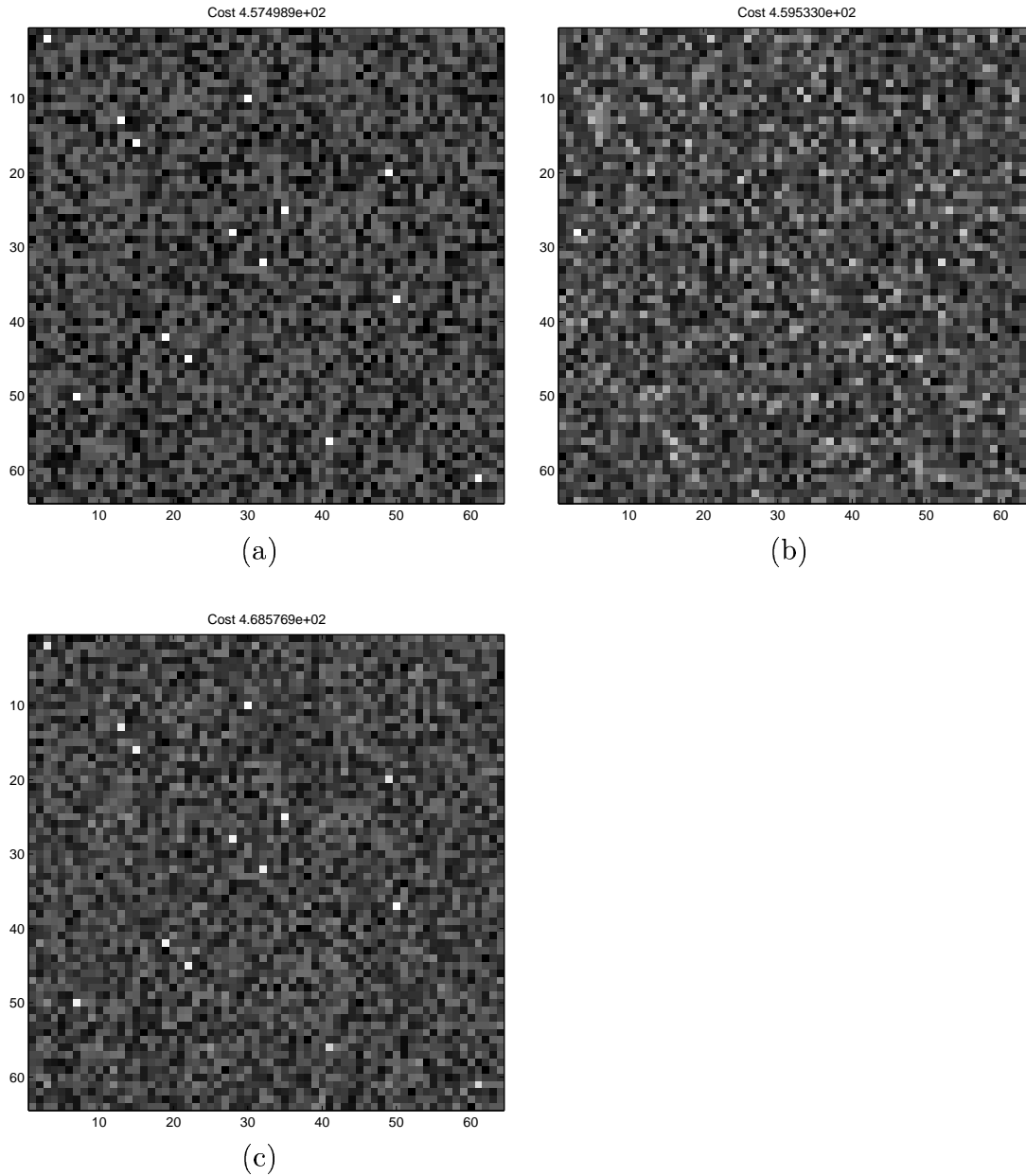


Figure 4.6 Entropy is not always a reliable measure of image quality. Entropy costs are expressed below, but they do not take into account the image normalization: (a) original image, with a cost of 4.57×10^2 , (b) SSA reconstruction, with a cost of 4.60×10^2 , and (c) image corrected with $\frac{\pi}{2}$ quantized phase error, with a cost of 4.69×10^2 .

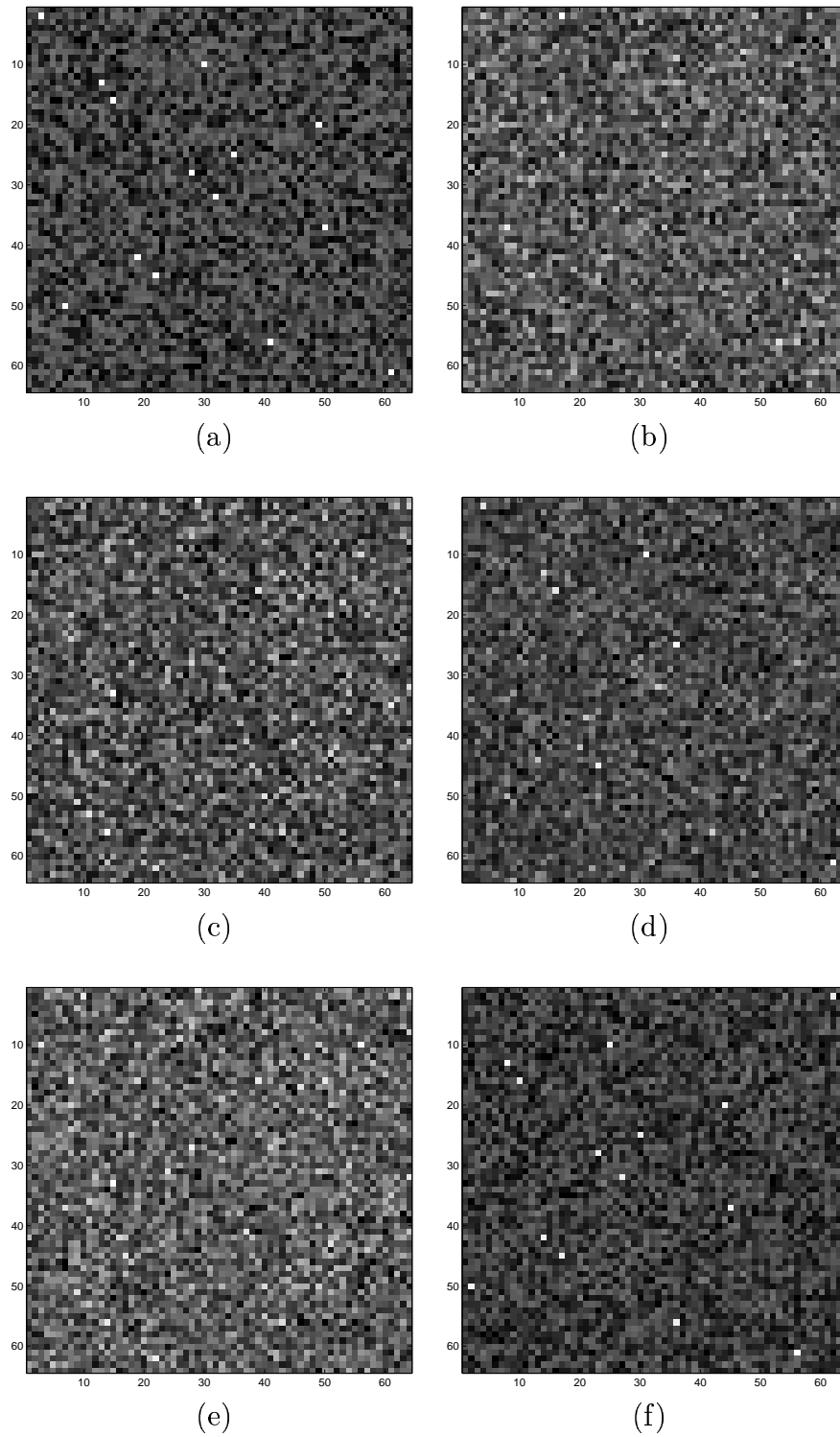


Figure 4.7 Synthesized test image (clutter range 0.35 with random phase in the spatial domain): (a) original image, (b) corrupt image, (c) standard SSA, (d) multi-dimensional SSA, (e) variable step size SSA, and (f) perturbation round SSA.

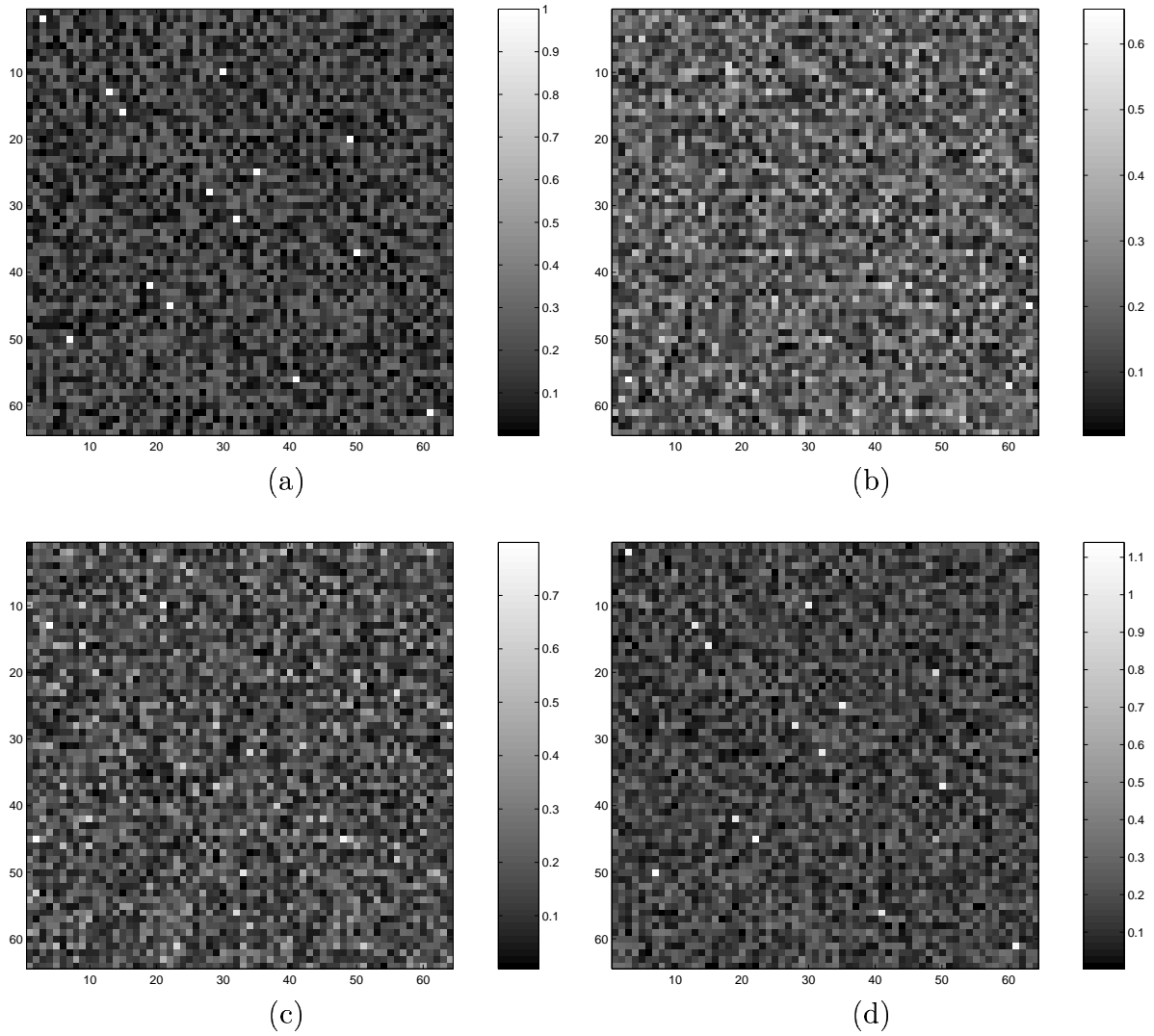


Figure 4.8 Synthesized test image (clutter range 0.35 with random phase in the spatial domain): (a) original image, (b) corrupted image, (c) standard SSA, and (d) modified simulated annealing.

CHAPTER 5

CONCLUSION

The stage-by-stage approaching (SSA) entropy minimization approach to synthetic aperture radar (SAR) autofocus is a modern and innovative technique, which to date has not been compared with other approaches. This thesis provides such a comparison and demonstrates that the entropy-based SSA is a promising autofocus method. The merit of the algorithm was evaluated through comparison with the established phase gradient autofocus (PGA) algorithm on both simulated and actual SAR data. Expediting implementation details for the SSA algorithm were presented. In addition, variations of the SSA algorithm aimed at improving its performance and computational expense were explored.

In the future, we hope to develop a theoretical analysis of the SSA approach. We seek to determine, using a formal proof, how well SSA can reconstruct a particular image. To accomplish this, we must first examine simple synthesized images and assume a particular model for the background clutter and phase error. The quality measure for how well SSA performs could be expressed in terms of the signal-to-clutter ratio of the reconstructed image, the mean-squared error (assuming that the reconstruction has been circularly shifted in place with respect to the original image), or the expected value of the magnitude of point scatterers. The proof would establish a bound on the performance of SSA, such that a particular degree of quality could be guaranteed under a specific model. In general, most of the autofocus literature does not present formal proofs of

performance, but rather provides qualitative examples and quantitative results based on simulations.

Another area of interest is the analysis of cost functions. We would like to determine the appropriateness of a cost function for a particular class of images. In addition, we wish to determine how prior knowledge of an uncorrupted image can be used to form a better reconstruction. The prior knowledge may include the number of point scatterers on a particular row, or a histogram.

In this thesis, we have examined several modifications to the SSA search strategy aimed at improving performance and reducing computational expense. We plan to explore the use of different optimization techniques to determine the entropy minimum. These techniques may include use of MATLAB's `fminu` nonlinear optimization routine, the Levenberg and Marquardt optimizer, or the method of conjugate gradients [12]. In the process of trying out different optimization methods, the effectiveness of the SSA search strategy may be better determined. In addition, new search techniques may be developed through these investigations.

It is also of interest to determine how seemingly different autofocus methods are related. It would be worthwhile to explore a possible unified autofocus theory. For instance, do PGA and SSA focus images in the same way, but using different steps? In addition, tools from related disciplines could be applied to the autofocus problem. Examples include maximum-likelihood deconvolution, adaptive filtering, and phase retrieval.

SAR autofocus for monostatic imaging scenarios has been considered to a great extent in the published literature. However, autofocus for multistatic radar systems remains a relatively unexplored area. A case of particular interest is passive radar, where transmissions from multiple radio and television stations are used to illuminate airborne targets. Passive radar is desired for covert detection and imaging, since active systems transmit detectable pulses. However, this is not attractive from an imaging standpoint because commercial stations transmit low bandwidth signals, as opposed to the specially crafted high bandwidth pulses of active systems. Consequently, only thin "arcs" of Fourier data are available instead of the 2-D polar annulus obtained using active SAR. Phase errors re-

sult from not precisely knowing the trajectory of the aircraft being imaged. Commercial systems for passively tracking aircraft are available, but their position estimates typically have errors of many wavelengths of the illuminating signal. Thus, they provide only a crude initial estimate. The autofocus problem becomes significantly more difficult in the passive case for two reasons. First, the signal structure has been reduced due to a limited amount of Fourier data. Second, the autofocus problem becomes two-dimensional; the phase error is no longer a function of the cross-range dimension only. It would be worthwhile to explore whether existing autofocus techniques could be applied to the passive scenario.

REFERENCES

- [1] C. V. Jakowatz, Jr., D. E. Wahl, P. H. Eichel, D. C. Ghiglia, and P. A. Thompson, *Spotlight-Mode Synthetic Aperture Radar: A Signal Processing Approach*. Boston: Kluwer Academic Publishers, 1996.
- [2] P. H. Eichel, D. C. Ghiglia, and C. V. Jakowatz, Jr., “Speckle processing method for synthetic aperture radar phase correction,” *Optics Letters*, vol. 14, pp. 1101–1103, January 1989.
- [3] F. Berizzi and G. Corsini, “Autofocusing of inverse synthetic aperture radar images using contrast optimization,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, no. 3, pp. 1185–1191, July 1996.
- [4] F. Berizzi, G. Corsini, M. Diani, and M. Veltroni, “Autofocusing of wide azimuth angle SAR images by contrast optimisation,” in *International Geoscience and Remote Sensing Symposium*, 1996, pp. 1230–1232.
- [5] L. Xi, L. Guosui, and J. Ni, “Autofocusing of ISAR images based on entropy minimization,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 35, no. 4, pp. 1240–1252, October 1999.
- [6] D. C. Munson, Jr., J. D. O’Brien, and W. K. Jenkins, “A tomographic formulation of spotlight-mode synthetic aperture radar,” *Proceedings of the IEEE*, vol. 71, no. 8, pp. 917–925, August 1983.
- [7] D. E. Wahl, P. H. Eichel, D. C. Ghiglia, and C. V. Jakowatz, Jr., “Phase gradient autofocus—a robust tool for high resolution SAR phase correction,” *IEEE*

Transactions on Aerospace and Electronic Systems, vol. 30, no. 3, pp. 827–835, July 1994.

- [8] J. W. Goodman, *Statistical Optics*. New York: John Wiley and Sons, 1985.
- [9] J. M. Brokish (private communication), 2002.
- [10] J. R. Fienup and J. J. Miller, “Generalized image sharpness metrics for correcting phase errors,” presented at Signal Recovery and Synthesis Topical Meeting of the Optical Society of America, Albuquerque, NM, 2001.
- [11] A. Y. Zomaya, “Natural and simulated annealing,” *Computing in Science and Engineering*, vol. 3, no. 6, pp. 97–99, November/December 2001.
- [12] J. R. Fienup, “Synthetic-aperture radar autofocus by maximizing sharpness,” *Optics Letters*, vol. 25, no. 4, pp. 221–223, February 2000.