Monotonic Iterative Algorithm for Minimum-Entropy Autofocus

Thomas Kragh

14th Annual ASAP Workshop

6-7 June 2006
Outline

• SAR motion blurring
• Minimum-Entropy SAR autofocus
• Fast monotonic algorithm
• Results using LIMIT SAR imagery
• Summary
SAR Image

Ideal Linear Synthetic Aperture

Slant Range

Cross Range
SAR Image with Simulated Motion Error (Before Autofocus)
SAR Image with Simulated Motion Error (After Autofocus)

Perturbed Synthetic Aperture

Pulse-by-Pulse Correction

Slant Range

Cross Range
SAR Motion Error Model

- To first order, motion errors appear as phase errors in the pulse-history

- Compensate by applying a pulse-by-pulse phase correction \( \phi = (\phi_1, \ldots, \phi_N) \) to the pulse history

- Focused image is given by the 1D inverse DFT along the phase corrected pulse history

- **Autofocus Problem**: What are the phase corrections?
Phase Gradient Autofocus (PGA)

Input Image → Find Brightest Points → Center → Data Matrix

FFT → Phase Correction → IFFT

\[ e^{j\phi_k} = \frac{v_k}{|v_k|} \]

\[ R = Z^H Z \]

\[ v = \text{top eigenvector} \]

- What is this optimizing?
- When do we stop iterating?
- Does it even converge?
Outline

• SAR motion blurring

• Minimum-Entropy SAR autofocus

• Fast monotonic algorithm

• Results using LIMIT SAR imagery

• Summary
For an image with complex-valued pixels $z_{mn} \in \mathbb{C}$, the image entropy is defined as

$$S = - \sum_{m,n} |z_{mn}|^2 \log(|z_{mn}|^2)$$

where

$$\sum_{m,n} |z_{mn}|^2 = 1$$

### Examples of Image Entropy

- $S = 0$
- $S = 10.585$
- $S = 11.245$
- $S = 11.625$
Gradient-Based Minimum-Entropy Autofocus

\[\phi^{(l)}\] 
\[e^{j\phi^{(l)}}\] 
\[\phi^{(0)}\] 
\[S, \nabla S\] 
\[\text{Calculate Image Entropy and Gradient}\] 
\[\text{Input Image}\] 
\[\text{Output Image}\] 

\[\text{FFT}\] 
\[\times\] 
\[\text{IFFT}\] 

\[\text{Phase Estimate at } l^{th}\text{-iteration}\] 
\[\text{Pulse-by-Pulse Phase Correction}\] 
\[\text{Monotonic Numerical Minimizer}\] 

Iterate until converged

Initial Phase Estimate

Image Entropy and Gradient
## Comparison of Autofocus Techniques

<table>
<thead>
<tr>
<th></th>
<th>Phase Gradient Autofocus (PGA)</th>
<th>Minimum Entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Modeling Assumptions</strong></td>
<td>Isolated point-sources</td>
<td>Non-Gaussian pixel distribution</td>
</tr>
<tr>
<td><strong>Theoretical Foundation</strong></td>
<td>Approximately maximum-likelihood phase estimation</td>
<td>Optimize image focus</td>
</tr>
<tr>
<td><strong>Limitations</strong></td>
<td>Closely spaced point-sources</td>
<td>Traditionally been computationally expensive</td>
</tr>
<tr>
<td></td>
<td>Low SNR</td>
<td></td>
</tr>
<tr>
<td><strong>Iterative</strong></td>
<td>Potentially</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Outline

• SAR motion blurring

• Minimum-Entropy SAR autofocus

• Fast monotonic algorithm

• Results using LIMIT SAR imagery

• Summary
Example Newton-Raphson Iteration

\[ \Phi(x) = \ln(x + 6) + \frac{12}{x + 6} \]
Example Newton-Raphson Iteration

2nd-order Taylor series

\( \Phi(n) \)
\( \Phi(n+1) \)

\( x^{(n)} \) \( x^{(n+1)} \)

step-size

decrease in objective
Newton-Raphson Counter-Example: Divergent Behavior
Newton-Raphson Counter-Example: Divergent Behavior

NR algorithm diverges when curvature is negative

2nd-order Taylor series
Example Monotonic Iteration

![Graph showing a monotonic relationship between two variables. The graph has a horizontal axis labeled 'x' ranging from 0 to 25, and a vertical axis ranging from 3.4 to 4. The curve starts high on the left, decreases, and then increases as it moves to the right.]
Example Monotonic Iteration

Monotonic algorithm decreases objective at each iteration.

\[ \Phi^{(n)} \]
\[ \Phi^{(n+1)} \]

Monotonic surrogate function

Decrease in objective

Step-size

\[ x^{(n+1)} \]
\[ x^{(n)} \]
Optimization Transfer

• Transfer a hard problem into a sequence of easy problems
  – Form an upper-bound at current estimate
    \[ \Phi(x) \leq \Theta(x; x^{(n)}) \]
  – Minimize upper-bound to get next estimate
    \[ x^{(n+1)} = \arg \min_x \Theta(x; x^{(n)}) \]
  – Converges to desired minimum if sequence of upper-bounds satisfy a few conditions
    \[ \lim_{n \to \infty} x^{(n)} = \arg \min_x \Phi(x) \]

• Results in iterative algorithm “fine-tuned” to our objective
Monotonicity Conditions

- **Monotonicity Condition**
  - change in the objective function is bounded above at each iteration
  \[
  \Phi(x) - \Phi(x^{(n)}) \leq \Theta(x; x^{(n)}) - \Theta(x^{(n)}; x^{(n)}) \leq 0 \quad \forall x
  \]

- **Following three conditions are sufficient for a differentiable surrogate**
  - Matches current value
    \[
    \Theta(x^{(n)}; x^{(n)}) = \Phi(x^{(n)})
    \]
  - Matches gradient
    \[
    \nabla \Theta(x; x^{(n)}) \bigg|_{x^{(n)}} = \nabla \Phi(x) \bigg|_{x^{(n)}},
    \]
  - Lies above
    \[
    \Theta(x; x^{(n)}) \geq \Phi(x) \quad \forall x
    \]

- **Will attain local minimum of \( \Phi \), global minimum if unique.**
Surrogate Function for Image Entropy

\[ S(\phi) = -\sum_{m,n} |z_{mn}(\phi)|^2 \log(|z_{mn}(\phi)|^2) \leq -\sum_{m,n} |z_{mn}(\phi)|^2 \log(|z_{mn}(\phi^{(l)})|^2) \]

- Satisfies Monotonicity Conditions
  - Lies above objective function for all phase values \( \phi \)
  - Tangent to objective function at current iterate \( \phi^{(l)} \)
Minimizing the Image Entropy Surrogate Function

- Image entropy surrogate function

$$\Theta(\phi; \phi^{(l)}) = - \sum_{m,n} |z_{mn}(\phi)|^2 \log(|z_{mn}(\phi^{(l)})|^2)$$

- Solve for next iterate by minimizing surrogate function

$$\phi^{(l+1)} = \arg\min_{\phi} \Theta(\phi; \phi^{(l)})$$

  - Difficult due to coupling of phase parameters in intensity

- Solution: Coordinate Decent approach
  - Minimize one parameter at a time while holding others fixed, cycle through all parameters

  - Minimizer has a simple closed-form solution!
Coordinate Descent using Surrogate Functions

Image Entropy

\( \phi_k \) (radians)

\( \phi_{k+1} \) (radians)

Image Entropy

\( \phi_k \) (radians)

\( \phi_{k+1} \) (radians)

Entropy
Surrogate

9.74
9.73
9.72
9.71
9.7
9.705
9.70
9.715
9.72
9.725
9.73
9.74
Given: \( y_{mn}, \phi^{(0)} \)

\[
\begin{align*}
\tilde{y}_{mk} &= \text{DFT}_n[y_{mn}] \\
z^{(0)}_{mn} &= \text{DFT}_k^{-1}[e^{j\phi^{(0)}_k} \tilde{y}_{mk}]
\end{align*}
\]

while Change in Image Entropy > Tolerance

for \( k = 1, \ldots, N \)

\[
\begin{align*}
A_k &= -\partial^2 \Theta(\phi; \phi_k^{(l)}) / \partial \phi_k^2 \bigg|_{\phi = \phi^{(l)}} \\
B_k &= \partial \Theta(\phi; \phi_k^{(l)}) / \partial \phi_k \bigg|_{\phi = \phi^{(l)}} \\
\phi_k^{(l+1)} &= \phi_k^{(l)} + \tan^{-1}(B_k/A_k) \\
z_{mn}^{(l,k+1)} &= z_{mn}^{(l,k)} + \frac{1}{N} e^{j2\pi kn/N} \left( e^{j\phi_k^{(l+1)}} - e^{j\phi_k^{(l)}} \right) \tilde{y}_{mk}
\end{align*}
\]

end

\[
\begin{align*}
l &= l + 1 \\
S^{(l)} &= - \sum_{m,n} |z_{mn}^{(l)}|^2 \log(|z_{mn}^{(l)}|^2)
\end{align*}
\]

end

\(~ O(MN) / \text{iter} \)

\(~ O(MN^2) \)

\(~ O(MN) \)

\(~ O(MN) \)

\(~ O(MN) \)

\(~ O(MN) \)

\(~ O(MN) \)

\(~ O(MN) \)

\(~ O(MN) \)

\(~ O(MN) \)

M = Range Bins
N = Pulses
Simultaneous Update using Surrogate Functions

Image Entropy

\( \phi_k \) (radians)

\( \phi_{k+1} \) (radians)

Image Entropy

Entropy

Surrogate

9.74
9.73
9.72
9.71
9.7
9.725
9.720
9.715
9.710
9.705
9.7
9.72
9.73
9.74

-3
-2
-1
0
1
2
3
-3
-2
-1
0
1
2
3
Simultaneous Update Algorithm
Computational Complexity

Given:  \( y_{mn}, \phi^{(0)} \)

\[
\tilde{y}_{mk} = DFT_n[y_{mn}]
\]
\[
z_{mn}^{(0)} = DFT_k^{-1}[e^{i\phi_k^{(0)}} \tilde{y}_{mk}]
\]

while Change in Image Entropy > Tolerance

\[
A_k = -\frac{\partial^2 \Theta(\phi; \phi_k^{(l)})}{\partial \phi_k^2} \bigg|_{\phi=\phi^{(l)}}
\]
\[
B_k = \frac{\partial \Theta(\phi; \phi_k^{(l)})}{\partial \phi_k} \bigg|_{\phi=\phi^{(l)}}
\]
\[
\phi_k^{(l+1)} = \phi_k^{(l)} + \tan^{-1} \left( \frac{B_k}{A_k} \right)
\]
\[
z_{mn}^{(l+1)} = DFT_k^{-1}[e^{i\phi_k^{(l+1)}} \tilde{y}_{mk}]
\]
\[
l = l + 1
\]
\[
\Phi^{(l)} = -\sum_{m,n} |z_{mn}^{(l)}|^2 \log(|z_{mn}^{(l)}|^2)
\]

end

• Note: No longer guaranteed to be monotonic!

\( M = \text{Range Bins} \)
\( N = \text{Pulses} \)
Outline

- SAR motion blurring
- Minimum-Entropy SAR autofocus
- Fast monotonic algorithm
- Results using LIMIT SAR imagery
- Summary
Results for Low-Order Phase Error

Before Autofocus

After Autofocus

2.4GHz AMD Opteron
- 9.3 min (40 s/iter)
- 2.9 s (0.6 s/iter)

True Phase Error
PGA (5 iterations)
Minimum Entropy
Results for Low-Order Phase Error

Before Autofocus

After Autofocus

2.4GHz AMD Opteron

2.9 s (0.6 s/iter)

3.46 s (0.18 s/iter)

Monotonic
TJK 7/11/2006
Results for High-Order Phase Error

Before Autofocus

After Autofocus

2.9 s (0.6 s/iter)

3.0 s (0.18 s/iter)

2.4GHz AMD Opteron
Conclusions

• Image entropy based autofocus is competitive with PGA
  – Equivalent computational complexity of ~O(MN log N)

• Advantages of image-entropy based autofocus
  – Directly optimize a measure of image quality
  – Guaranteed convergence for sequential Coordinate Descent algorithm
  – “For all practical purposes” convergence for Simultaneous Update

• Disadvantages of image-entropy based autofocus
  – Slower initial convergence rate than PGA
  – No convergence guarantee for Simultaneous Update algorithm

• Best of both worlds
  – Initialize with PGA solution, then iterate using Image Entropy
  – Monotonicity guarantees an improved estimate every iteration
Summary

- Developed an iterative minimum-entropy autofocus algorithm with guaranteed monotonic convergence

- Presented an efficient $\sim O(MN \log N)$ implementation that is computationally competitive with PGA

- Demonstrated on realistic SAR imagery
Backup
Examples of Image Entropy

\[ \Phi = 0 \]

\[ \Phi = \log(3) \]

\[ \Phi = \log(7) \]

\[ \Phi = 10.585 \]

\[ \Phi = 11.245 \]

\[ \Phi = 11.625 \]
Invariance Properties of Image Entropy

Scale Invariance

Permutation Invariance

• Relevance to AutoFocus
  – Invariant to constant phase error
  – Invariant to linear phase error (image shift)
Surrogate Function Curvature and Convergence Rate

- For fastest convergence rate, find the smallest curvature surrogate that does not violate the monotonicity conditions.
Coordinate Decent Algorithm

• Let $\phi^{(l,k)} = \{\phi^{(l+1)}_1, \ldots, \phi^{(l+1)}_{k-1}, \phi^{(l)}_k, \phi^{(l)}_{k+1}, \ldots, \phi^{(l)}_N\}$ be the phase estimate at the (l)th iteration, where the first (k-1) parameters have already been updated.

• Similarly, let $\phi = \{\phi^{(l+1)}_1, \ldots, \phi^{(l+1)}_{k-1}, \phi^{(l)}_k, \phi^{(l)}_{k+1}, \ldots, \phi^{(l)}_N\}$ be our free parameter to minimize over.

• Our surrogate then reduces to the following scalar function,

$$\Theta(\phi; \phi^{(l,k)}) = \psi_k(\phi_k)$$

and the joint minimization reduces to a scalar minimization,

$$\phi^{(l+1)}_k = \arg\min_{\phi_k} \psi_k(\phi_k)$$
Coordinate Decent Algorithm (cont.)

• Scalar surrogate satisfies the linear ODE \( \psi' = -\psi''' \)
  thus has a solution of the form

\[
\psi(\phi) = A_k \cos(\phi - \phi_k^{(l)}) + B_k \sin(\phi - \phi_k^{(l)}) + C_k
\]

where the relevant constants are given by

\[
A_k = -\psi''(\phi_k^{(l)}) \quad \text{and} \quad B_k = \psi'(\phi_k^{(l)})
\]

• Scalar surrogate function can be minimized in closed form

\[
\phi_k^{(l+1)} = \arg \min_{\phi} \psi(\phi) = \phi_k^{(l)} + \tan^{-1}\left(\frac{B_k}{A_k}\right)
\]