A butterfly algorithm for synthetic aperture radar

A Butterfly Algorithm for Synthetic Aperture Radar

Laurent Demanet[a] Matthew Ferrara[b] Nicholas Maxwell[c] Jack Poulson[d] Lexing Ying[e]

[a]Massachusetts Institute of Technology, Department of Mathematics, Cambridge, MA
[b]Matrix Research, Inc., Dayton, OH
[c]University of Houston, Department of Mathematics, Houston, TX
[d]University of Texas at Austin, ICES, Austin, TX
[e]University of Texas at Austin, Department of Mathematics and ICES, Austin, TX

ABSTRACT

It is not currently known if it is possible to accurately form a synthetic aperture radar image from N data points in provable near-linear complexity, where accuracy is defined as the ℓ_2 error between the full $O(N^2)$ backprojection image and the approximate image. To bridge this gap, we present a backprojection algorithm with complexity O(log(1/ϵ)N log N), with ϵ the tunable pixelwise accuracy. It is based on the butterfly scheme, which works for vastly more general oscillatory integrals than the discrete Fourier transform. Unlike previous methods this algorithm allows the user to directly choose the amount of acceptable image error based on a well-defined metric. Additionally, the algorithm does not invoke the far-field approximation or place restrictions on the antenna flight path, nor does it impose the frequency-independent beampattern approximation required by time-domain backprojection techniques.

1. INTRODUCTION

1.1 Setup

Synthetic-aperture radar (SAR) is an imaging modality that produces images of a scene from measurements of scattered electromagnetic waves. Pulses of microwaves are sent from an antenna aboard an airplane or a satellite, scattered by objects on the surface of the Earth, and recorded by the same (or a different) antenna. The imaging problem consists in recovering a reflectivity profile that explains the recorded pulse-echo data.

- *Image space* is indexed by $\mathbf{x} = (x, y) \in R^2$, the horizontal coordinates. The scatterers are assumed to be at a known elevation $z = h(x, y)$, so we have the embedding $\mathbf{x}_T = ((x, y), h(x, y)) \in \mathbb{R}^3$. The reflectivity profile is a function $m(\mathbf{x})$ whose magnitude indicates the strength of the reflection by the object at x_T, as an impedance contrast for instance.
- *Data space* is indexed by ω, the frequency of the recorded signal, and s, a parameter that defines the position of the antenna through a function $\gamma(s) \in \mathbb{R}^3$. Data are given by a function $d(\omega, s)$, whose value is the result of a measurement of the strength of the recorded signal at angular frequency $\omega = 2\pi f$, when the antenna is at $\gamma(s)$.

Under very general and widely accepted assumptions*, this imaging map is an oscillatory integral. We make three additional but unessential assumptions that can easily be removed: 1) monostatic SAR in which the transmitter antenna is also the receiver, 2) no consideration of the orientation of the antenna, and 3) the phase-center approximation, in which the antenna is far enough from the targets that it is considered as a point. Imaging is then done by some "generalized" filtered backprojection:

$$m(\mathbf{x}) = \int_\Omega e^{-2i\omega|\gamma(s)-\mathbf{x}_T|/c} B(\omega, s, \mathbf{x}) d(\omega, s)\, ds d\omega, \tag{1}$$

*Single scattering in the Born approximation, scalar wavefields, no dispersion, no attempt at addressing three-dimensional effects such as shadowing and layover, start-stop setup, no attempt at estimating target motion. This is the setup in Cheney and Borden.[12]

Algorithms for Synthetic Aperture Radar Imagery XVIII, edited by Edmund G. Zelnio, Frederick D. Garber,
Proc. of SPIE Vol. 8051, 805105 · © 2011 SPIE · CCC code: 0277-786X/11/$18 · doi: 10.1117/12.888948

where $B(\omega, s, \mathbf{x})$ is an amplitude function, and $\mathbf{x}_T = (x_1, x_2, h(x_1, x_2))$ is the target point. We will comment later on the backprojection interpretation. See Cheney and Borden[12] for the justification of this formula.

Here Ω is the acquisition manifold, normally a rectangle $[\omega_{\min}, \omega_{\max}] \times [s_1, s_2]$. The amplitude factor $B(\omega, s, x)$ is chosen so that the formula above is a good approximate inverse to the forward/modeling/reprojection operator

$$d(\omega, s) = \int e^{2i\omega|\gamma(s)-x_T|/c} A(\omega, s, x) m(x)\, dx_1 dx_2. \tag{2}$$

In this integral, the amplitude factor $A(\omega, s, x)$ is

$$A(\omega, s, x) = -\omega^2 P(\omega) \frac{J(\omega, \widehat{x_T - \gamma(s)}) W(\omega, \widehat{x_T - \gamma(s)})}{(4\pi|x_T - \gamma(s)|^2)},$$

where $P(\omega)$ is the transfer function of the pulse, and J and W are the respective antenna beam patterns at transmission and reception. The hat over a vector denotes unit length normalization. The corresponding amplitude B for imaging is cumbersome to write precisely without introducing the so-called Stolt change of variables; suffice it to say that

$$B = \frac{\chi}{A\, d_B},$$

where A is the amplitude above, d_B is the so-called Beylkin determinant, and χ is an ad-hoc cutoff that prevents division by zero.

The contribution of this paper is to propose a fast and accurate way of evaluating oscillatory integrals such as (1). We start by reviewing the existing algorithms and their range of applicability.

1.2 Existing algorithms

Denote by $\Delta\omega = \omega_{\max} - \omega_{\min}$ the bandwidth of the measurements. For simplicity, we will assume that the bandwidth is on the same order of magnitude as the representative "carrier" frequency $\omega_0 \simeq (\omega_{\min} + \omega_{\max})/2$, so we have broadband measurements.

The Nyquist-Shannon sampling rate should be respected both in image space and in data space.

- In image space, we expect variations on the order of the wavelength c/ω_0 in both directions[†], and the scene to be imaged has sidelength L, so the total number of pixels is proportional to $L^2\omega_0^2/c^2$.
- In data space, a frequency grid spacing of $O(c/L)$ is called for to access distances on the order of L, so we need $O(\omega_0 L/c)$ samples. The distance between pulses should be on the order of the wavelength $O(c/\omega_0)$ to attain the same wavelength resolution on the ground, so we need $O(\omega_0 L/c)$ samples in slow time as well. So the total number of data points is proportional to $L^2\omega_0^2/c^2$.

The complexity of specifying either a dataset, called N, is therefore proportional to the complexity of specifying an image, and increases quadratically in the frequency ω_0:

$$N = O(L^2\omega_0^2/c^2).$$

It is the scaling of the complexity of the imaging algorithm as a function of this parameter N which is of interest. We refer to direct summation as the naive algorithm, with complexity $O(N^2)$.

Traditionally, it is only in contexts where the problem formulation is *simplified* that designing faster algorithms is possible. Two levels of simplification are popular in the literature:

[†]This can be refined by considering range direction and cross-range direction, in the case of narrowband measurements.

1. The separability assumption $B(\omega, s, x) = P(\omega)Q(s, x)$. This assumption only makes sense if the antenna beam patterns are independent of frequency. In this setting, we may evaluate equation (1) by the following sequence of steps: for each s, multiply the data by $1/P(\omega)$, perform a Fourier transform in ω evaluated at $2|\gamma(s) - \mathbf{x}_T|/c$, and multiply by $Q(s, x)$. Iterate and sum over s. This procedure results in an algorithm of complexity $O(N^{3/2})$. It is called filtered backprojection (proper), because it can be seen as integration along circles of equal range when expressed as acting on data $\hat{d}(t, s)$ of time t. One would also speak of a generalized Radon transform.[3] Further work can go into simplifying the computation of the remaining sum over s to lower the overall complexity, like in the remarkable work of Ulander et al.[30] on multiscale "beamforming." Ulander obtained a $O(N \log N)$ algorithm, but within the further limitation of an omni-directional antenna, flat topography, and that deviations from a linear track are treated by a perturbation argument. It is also unclear that a pointwise or mean-square error estimate would hold for this algorithm.

2. The far-field assumption $\|\gamma(s) - x_T\| \simeq \|\gamma(s)\| - \widehat{x}_T \cdot \gamma(s)$. This simplification makes sense if the target is so far from the antenna that the circles of equal distance can be treated as straight lines. In this setting equation (1) becomes a 2D Fourier transform, albeit not on a uniform grid. In the time domain, we would speak of a Radon transform instead of a generalized Radon transform. The USFFT method of Dutt and Rokhlin[15] and its variants[4,8] apply to this problem and yield algorithms of complexity $O(N \log N)$. In this setting the Polar Format Algorithm (PFA),[31] which interpolates the data from polar raster onto a rectilinear grid, can be a reasonable approach. A comparison between PFA, USFFT, and NUFFT techniques for SAR is given in Andersson et al.[1] Backprojection algorithms of a different kind have also been designed in the time domain for this task: this includes work by Basu and Bresler,[2] and Nilsson,[22] which can be seen as special cases of Ulander's work.[30]

In contrast, this paper presents a fast "butterfly" algorithm useful for much more general radar setups. None of the assumptions above are made; only minimal smoothness properties of the functions $\gamma(s)$ and $B(\omega, s, x)$ are required. In fact, the butterfly scheme is intrinsically robust and we anticipate that it would easily accommodate refinements such as multistatic SAR (several sources and antennas), or taking into account the orientation of the antenna via the pitch, roll and yaw angles as measured by the Inertial Navigation System (INS).

The main idea behind the butterfly scheme is that of *low-rank interaction*. This idea conveys a very important and general principle of quantification of the "information content" in high-frequency scattering.

1.3 Low-rank interactions

The phase center of an antenna is the point $\gamma(s)$ introduced earlier, about which the antenna beam patterns are constructed as functions of angular frequency ω direction $\widehat{x - \gamma}$. It draws its name from the fact that a more accurate representation of the radiation field from an antenna Γ is (we drop the s dependence of γ for the time being.)

$$\begin{aligned} u(x, \omega) &= \int_\Gamma \frac{e^{ik|x-y|}}{4\pi|x-y|} j(y, \omega)\, dS_y, \\ &\simeq \frac{e^{ik|x-\gamma|}}{4\pi|x-\gamma|} \int_\Gamma e^{-ik(\widehat{x-\gamma})\cdot y} j(y, \omega)\, dS_y, \\ &=: \frac{e^{ik|x-\gamma|}}{4\pi|x-\gamma|} J(\omega, \widehat{x-\gamma}) \qquad\qquad (\omega = kc), \end{aligned}$$

hence γ should be chosen as a good "center" for the approximately circular phase lines of $u(x, \omega)$. Here $j(y, \omega)$ is a scalar counterpart of the current density on the antenna[‡]. To pass to the second line the well-known far-field approximation $|x-\gamma| \gg |y-\gamma|$ was used. While the full surface integral over the antenna is impractical for radar imaging, this phase center reduction has the advantage of presenting the antenna beam patterns as functions on the sphere of outgoing directions. (A similar argument can be made for the receiving antenna.)

[‡]We apologize in passing to the engineers who are used to $j = \sqrt{-1}$.

Another way of reducing the complexity of the integral, without making the far-field approximation, consists in finding several *equivalent sources* γ_i, with weights J_i; as well as several regions $\mathcal{A}$ such that

$$u(x,\omega) = \sum_i \frac{e^{ik|x-\gamma_i|}}{4\pi|x-\gamma_i|} J_i + O(\epsilon), \qquad x \in \mathcal{A}.$$

Here, the error is under control and denoted ϵ. In other words, if we are willing to restrict ourselves to a certain region $\mathcal{A}$ of space, how many "phase centers" γ_i indexed by i are really needed to synthesize the radiation field to prescribed accuracy? Ultimately, a low-order equivalent-source approximation is attainable because of the *low-rank factorization* property of the Green's function when y is restricted to the antenna and x is restricted to the box $\mathcal{A}$.

The underlying fundamental property of Green's functions is that factorization is guaranteed to work, with a low rank independent of ω, if the following adimensional number is low,

$$F = \frac{\mathrm{diam}(\Gamma) \times \mathrm{diam}(\mathcal{A})}{\lambda \times d(\Gamma, \mathcal{A})}.$$

We may call F an "algorithmic Fresnel number" in analogy with the discussion of Fraunhofer vs. Fresnel diffraction in physics textbooks. Its value should be comparable to 1 or lower for the low-rank property to hold. Here $\mathrm{diam}(\Gamma)$ is the antenna diameter, $\mathrm{diam}(\mathcal{A})$ is the largest diagonal of the box $\mathcal{A}$, $\lambda = 2\pi/\omega$ is the wavelength, and $d(\Gamma, \mathcal{A})$ is the distance between the antenna and the box. Similar ideas appear in work of Michielssen and Boag,[21] Engquist and Ying,[16] Candès, Demanet, and Ying,[10] Rokhlin,[24] Brandt,[6] and likely many others.

1.4 The butterfly algorithm

The butterfly algorithm is a systematic way of leveraging low-rank interactions in the scope of a fast algorithm for oscillatory integrals. The pulse-echo data now replaces the antenna as a virtual "source" of radiation, so the physical problem is different from that presented in the previous section, but the ideas remain the same.

The butterfly algorithm originates from work of Michielssen and Boag,[21] and has recently been popularized in a string of papers by Rokhlin and O'Neil,[23] Ying,[33] Candès, Demanet and Ying[10] and Tygert.[29] Note that the algorithmic variant presented in our earlier work[10] is particularly well suited for the application to SAR imaging: unlike ONeil and Rokhlin[23] it does not have a precomputation step. The butterfly is a natural descendant of, or variant of, the fast multipole method[18,24] for high-frequency scattering, in the sense that low-rank interactions are adequate "summaries" that serve as a substitute for multipole expansions.[34]

If we let $y = (\omega', s)$, with $\omega' = \omega/\omega_0$ a rescaled frequency variable, then we may succinctly write the imaging operation as

$$m(x) = \sum_y K(x,y) d(y), \tag{3}$$

with $K(x,y)$ the oscillatory kernel. Low-rank interactions come into play through the problem of finding a good approximation

$$m(x) = \sum_{j=1}^{r} K(x, y_j)\delta_j + O(\epsilon), \tag{4}$$

where (y_j, δ_j) are equivalent sources. In order for the number r of terms to be independent of the carrier frequency ω_0 (or $N \sim \omega_0^2$), it suffices to take $x \in A$, and to restrict the sum to $y \in B$, in such a way that the "algorithmic Fresnel number" is small, i.e.

$$\mathrm{diam}(A) \times \mathrm{diam}(B) \leq \frac{H}{\omega_0}, \tag{5}$$

for some constant H that has the dimension of a length, with value comparable to the altitude of the antenna. This property of r was established in earlier work of two of the authors, in the more general setting of Fourier integral operators.[10] It holds for SAR imaging if the trajectory $\gamma(s)$ is smooth, i.e., a C^∞ function of s.

If B would cover the whole data space, we would be in presence of the full sum. In that case the range of validity of a formula like (4) would be restricted to very small boxes A – of diameter $O(1/\omega_0)$ – and to each such small box A would correspond a given set of equivalent sources (y_j, δ_j). If the information of the (y_j, δ_j) were available for each small box A, then we would be in presence of a very fast algorithm: a superposition of $r = O(1)$ terms for each of the $O(\omega_0^2) = O(N)$ boxes in A would suffice for imaging. This is unfortunately not the case.

The butterfly scheme is a way of computing these equivalent sources by playing on the sizes of A and B in a multiscale manner. It is possible to tile model and data space with boxes that satisfy the scaling (5), and consider the low-rank interactions between any pair of such boxes. It is advantageous to generate such tilings by means of quadtree partitions of model and data space. See Figure 1.4, where data space (y) is on the right, and model space (x) on the left.

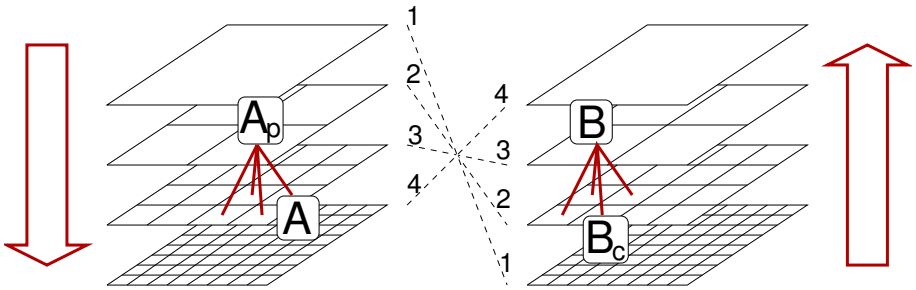

Figure 1. The two quadtrees of the butterfly scheme.

For instance,

- The fine boxes at the leaves (bottom) of the tree on the right can be paired with a large box at the root (top) of the tree on the left. The pairing corresponds to the dashed line labeled "1". If the boxes B are small enough ($1/\omega_0$ by $1/\omega_0$), then the scaling (5) is respected. This choice of tiling corresponds to sums (4) restricted to only a few terms: it is straightforward to compute directly, without the δ_j. But it is not very useful since we want the whole sum.
- On the other hand, the large box B at the root of the tree can be paired with small boxes A at the leaves. This pairing goes by the number "4". It corresponds to a low-rank view of the whole sum (3), only valid in certain very small sets A on the x-side. It is exactly what we are interested in, but the δ_j in the expansion are unknown to us.

The core of the butterfly algorithm is the ability to update low-rank interactions in a multiscale fashion, down the left tree and up the right tree, by respectively grouping and subdividing boxes. In the picture this allows to iteratively obtain the δ_j at all scales, from the pairing "1" to the pairing "4".

The details of the butterfly scheme concern the choice of y_j in (4), how to realize the low-rank expansion as an interpolation problem, and how to update the δ_j weights from one scale to the next. These details are presented in Section 2, and follow from our previous work in Candès, Demanet, and Ying.[10] Let us mention that it is the "Chebyshev interpolation" version of the butterfly algorithm which is used in this paper; it is unclear that the other variants would be equally well suited for SAR imaging.

We now state the rigorous performance guarantee enjoyed by the butterfly algorithm, which was missing in our previous work.[10]

1.5 Accuracy and complexity bounds

In this paper, like in Candès, Demanet, and Ying,[9,10] we choose the radius of convergence of Taylor expansions as a measure of smoothness of real-analytic functions. In one spatial dimension, a function $f(x)$ is (Q, R)-analytic if it is infinitely differentiable and its derivatives obey

$$|f^{(n)}(x)| \leq Q\, n!\, R^{-n}.$$

The number R is simply a lower bound on the radius of convergence of Taylor expansions of f, uniform over all points where f is considered. We say that a function $f(x)$ of $x \in \mathbb{R}^2$ is (Q, R)-analytic if its directional derivative along any line obeys the same estimate: for any unit-length d,

$$|(d \cdot \nabla)^n f(x)| \leq Q\, n!\, R^{-n}.$$

Our main assumption on the kernel $K(x, y)$ is that it can be written in Fourier integral form as $a(x, y)e^{iM\phi(x,y)}$, with the amplitude a and the phase ϕ both analytic in x and y separately. Manifestly, the SAR kernel of equation (1) is of the form $ae^{iM\phi}$ with $M = \omega_0 = O(\sqrt{N})$.

The following complexity result depends on N and ϵ and its dependence on Q and R will not be explicitly calculated.

THEOREM 1.1. *Assume that the flight path $\gamma(s)$ and the amplitude $B(\omega, x, s)$ are both real-analytic functions of their arguments. Write $y = (\omega/\omega_0, s)$, and*

$$K(x, y) = a(x, y)e^{iM\Phi(x,y)}.$$

Then the variant of the butterfly method presented in this paper, which uses Chebyshev interpolation, provides an approximation $\tilde{u}(x) = \sum \tilde{K}(x, y_j)f_j$ obeying

$$\|\tilde{u} - u\|_\infty \leq \epsilon \sum_j |f_j|,$$

in exact arithmetic, and in (sequential) algorithmic complexity

$$C(Q, R) \times \max\{\log^4(\frac{1}{\epsilon}),\ (\log^4 N)\ \log^4(C \log N)\} \times N \log N.$$

The proof will be supplied in a forthcoming paper. Note that the above theorem contains no statement about the discretization error; only the ℓ_∞ discrepancy between the result of naive summation and the result of the fast algorithm is controlled.

2. THE BUTTERFLY ALGORITHM FOR OSCILLATORY INTEGRALS

Let us denote by X the set of all x (positions) indexing model space, and by Y the set of of all y (normalized frequencies and slow times) indexing data space. From the discussion above, it is clear that both $|X|$ and $|Y|$ are on the order of $N = O(M^2)$. By rescaling the geometry if necessary, we can assume that X and Y are both supported in the unit square $[0, 1]^2$. In this section, unlike in the numerical code, we do not worry about the values of numerical constants: for brevety only the asymptotic behavior in terms of M is retained. The computational problem is then to approximate $u(x)$ defined by

$$u(x) = \sum_{y \in Y} a(x, y)e^{iM\Phi(x,y)} f(y).$$

We now give a brief discussion of the butterfly algorithm for computing this oscillatory summation. The presentation follows closely to the one of Candès, Demanet, and Ying[10] and the new contribution is an easy way to address the amplitude function $a(x, y)$.

Suppose that A and B are two square boxes in $[0, 1]^2$, while A is considered to be a box in the X domain and B in the Y domain. We denote their centers, respectively, by $x_0(A)$ and $y_0(B)$; and the length of their diagonals, respectively, by $\mathrm{diam}(A)$ and $\mathrm{diam}(B)$. The most important component of the butterfly algorithm is the existence of a low-rank approximation for the kernel

$$\left| a(x, y)e^{iM\Phi(x,y)} - \sum_{t=1}^{r_\epsilon} \alpha_t^{AB}(x)\beta_t^{AB}(y) \right| \leq \epsilon \tag{6}$$

for $x \in A$ and $y \in B$ when $\mathrm{diam}(A)\mathrm{diam}(B) \lesssim 1/M$. Define $u^B(x)$ to be the partial sum restricted to, or "potential generated by", $y \in B$. The benefit of the low-rank approximation is that it gives rise to a compact representation for $u^B(x)$ when restricted to $x \in A$:

$$u^B(x) \simeq \sum_{t=1}^{r_\epsilon} \alpha_t^{AB}(x) \left(\sum_{y \in B} \beta_t^{AB}(y) f(y) \right) \quad \forall x \in A.$$

Therefore, any coefficients $\{\delta_t^{AB}\}_t$ obeying

$$\delta_t^{AB} \simeq \sum_{y \in B} \beta_t^{AB}(y) f(y), \tag{7}$$

offer a good approximation to $u^B(x)$ for $x \in A$.

In order to find a low-rank approximation, we introduce the so-called residue phase associated with the pair (A, B)

$$R^{AB}(x, y) := \Phi(x, y) - \Phi(x_0(A), y) - \Phi(x, y_0(B)) + \Phi(x_0(A), y_0(B)), \tag{8}$$

Under the condition that $\Phi(x, y)$ is real-analytic both in x and in y, and $\mathrm{diam}(A)\mathrm{diam}(B) \lesssim 1/M$, it is easy to show that $R^{AB}(x, y) = O(1/M)$ for $x \in A$ and $y \in B$. As a result, it was shown in[10] that r_ϵ in equation (6) can be bounded by a constant times $\log^4(1/\epsilon)$. This bound can be further refined to a constant times $\log^2(1/\epsilon)$. The point is that those bounds on r_ϵ are *independent of* M, and only depend weakly on the desired accuracy.

One way to realize such low values of r_ϵ, as explained in Candès, Demanet, and Ying,[10] is to use polynomial interpolation in x when $\mathrm{diam}(A) \lesssim 1/\sqrt{M}$ and in y when $\mathrm{diam}(B) \lesssim 1/\sqrt{M}$. The interpolation points are placed on tensor Chebyshev grids for efficiency. For some small positive integer q, the Chebyshev grid of order q on the centered unit interval $[-1/2, 1/2]$ is defined by

$$\left\{ z_j = \frac{1}{2} \cos\left(\frac{j\pi}{q-1} \right) \right\}_{0 \le j \le q-1}.$$

The Lagrange basis polynomials $L_i(z)$ of this grid are given by

$$L_j(z) := \prod_{0 \le k \le q-1, k \neq j} \frac{z - z_k}{z_j - z_k}.$$

By taking tensor products, we can define the two dimensional Chebyshev grid $\{(z_{t_1}, z_{t_2})\}$ for the centered unit square and its Chebyshev basis functions

$$L_t(z_1, z_2) := L_{t_1}(z_1) \cdot L_{t_2}(z_2), \quad \text{for} \quad t = (t_1, t_2).$$

For a general square box B in the Y domain, its Chebyshev grid can be defined similarly by appropriate scaling and shifting. We denote this grid by $\{y_t^B\}$ and its Lagrange basis functions for its Chebyshev grid by $\{L_t^B\}$. When $\mathrm{diam}(B) \lesssim 1/\sqrt{M}$, Lagrange interpolation on the grid adapted to B provides the approximation

$$a(x, y) e^{iMR^{AB}(x,y)} \simeq \sum_t a(x, y_t^B) e^{iMR^{AB}(x, y_t^B)} L_t^B(y).$$

Similarly, for a box A in the X domain, its Chebyshev grid and Lagrange basis functions are denoted by $\{x_t^A\}$ and $\{L_t^A\}$, respectively. When $\mathrm{diam}(A) \lesssim 1/\sqrt{M}$, Lagrange interpolation on the grid adapted to A provides the approximation

$$a(x, y) e^{iMR^{AB}(x,y)} \simeq \sum_t L_t^A(x) a(x_t^A, y) e^{iMR^{AB}(x_t^A, y)}$$

The number q of Chebyshev points grows logarithmically in the error level ϵ, resulting in $r_\epsilon = q^2 = O(\log^2 1/\epsilon)$ as announced earlier. The section on numerical experiments contains more details on the choice of q vs. accuracy.

In practice, it is advantageous to take q with values ranging from 5 to 10 in order to obtain "a few" to "several" digits of accuracy.

To pass from low-rank approximations of $a(x,y)e^{iMR^{AB}(x,y)}$ to those for the true kernel $a(x,y)e^{iM\Phi(x,y)}$, we restore the other factors in (8). When $diam(B) \lesssim 1/\sqrt{M}$, this gives

$$a(x,y)e^{iM\Phi(x,y)} \simeq \sum_t \left(a(x,y_t^B)e^{iM\Phi(x,y_t^B)}\right)\left(e^{-iM\Phi(x_0(A),y_t^B)}L_t^B(y)e^{iM\Phi(x_0(A),y)}\right)$$

In terms of the earlier notations,

$$\alpha_t^{AB}(x) = a(x,y_t^B)e^{iM\Phi(x,y_t^B)}, \quad \beta_t^{AB}(y) = e^{-iM\Phi(x_0(A),y_t^B)}L_t^B(y)e^{iM\Phi(x_0(A),y)}, \tag{9}$$

and the expansion coefficients $\{\delta_t^{AB}\}_t$ for the potential should obey the condition

$$\delta_t^{AB} \simeq \sum_{y\in B}\beta_t^{AB}(y)f(y) = e^{-iM\Phi(x_0(A),y_t^B)}\sum_{y\in B}\left(L_t^B(y)e^{iM\Phi(x_0(A),y)}f(y)\right). \tag{10}$$

Similarly when $\mathrm{diam}(A) \lesssim 1/\sqrt{M}$, we have

$$a(x,y)e^{iM\Phi(x,y)} \simeq \sum_t \left(e^{iM\Phi(x,y_0(B))}L_t^A(x)e^{-iM\Phi(x_t^A,y_0(B))}\right)\left(a(x_t^A,y)e^{iM\Phi(x_t^A,y)}\right)$$

In terms of the earlier notations,

$$\alpha_t^{AB}(x) = e^{iM\Phi(x,y_0(B))}L_t^A(x)e^{-iM\Phi(x_t^A,y_0(B))}, \quad \beta_t^{AB}(y) = a(x_t^A,y)e^{iM\Phi(x_t^A,y)}, \tag{11}$$

and the expansion coefficients $\{\delta_t^{AB}\}$ should obey

$$\delta_t^{AB} \simeq \sum_{y\in B}\beta_t^{AB}(y)f(y) = \sum_{y\in B}a(x_t^A,y)e^{iM\Phi(x_t^A,y)}f(y) = u^B(x_t^A). \tag{12}$$

Combining these expansions with the general structure of the butterfly scheme, we arrive at the following algorithm. It is a slight modification of the one proposed in Candès, Demanet, and Ying.[10]

1. *Preliminaries.* Construct two quadtrees T_X and T_Y for X and Y. Each leaf node of T_X and T_Y is of size (a constant times) $1/M \times 1/M$. We denote the number of levels of T_X and T_Y by L.

2. *Initialization.* Set A to be the root of T_X. For each leaf box $B \in T_Y$, construct the expansion coefficients $\{\delta_t^{AB}, 1 \le t \le r_\epsilon\}$ from (10) by setting

$$\delta_t^{AB} = e^{-iM\Phi(x_0(A),y_t^B)}\sum_{y\in B}\left(L_t^B(y)e^{iM\Phi(x_0(A),y)}f(y)\right). \tag{13}$$

3. *Recursion.* For each $\ell = 1, 2, \ldots, L/2$, construct the coefficients $\{\delta_t^{AB}, 1 \le t \le r_\epsilon\}$ for each pair (A,B) with A at level ℓ and B at the complementary level $L-\ell$ as follows: let A_p be A's parent and $\{B_c, c = 1,2,3,4\}$ be B's children. For each child, we have available from the previous level an approximation of the form

$$u^{B_c}(x) \simeq \sum_{t'} e^{iM\Phi(x,y_{t'}^{B_c})}\delta_{t'}^{A_pB_c}, \quad \forall x \in A_p.$$

Summing over all children gives

$$u^B(x) \simeq \sum_c\sum_{t'} e^{iM\Phi(x,y_{t'}^{B_c})}\delta_{t'}^{A_pB_c}, \quad \forall x \in A_p.$$

Since $A \subset A_p$, this is also true for any $x \in A$. This means that we can treat $\{\delta_{t'}^{A_pB_c}\}$ as equivalent sources in B. As explained below, we then set the coefficients $\{\delta_t^{AB}\}_t$ as

$$\delta_t^{AB} = e^{-iM\Phi(x_0(A),y_t^B)}\sum_c\sum_{t'} L_t^B(y_{t'}^{B_c})e^{iM\Phi(x_0(A),y_{t'}^{B_c})}\,\delta_{t'}^{A_pB_c}. \tag{14}$$

4. *Switch.* The interpolant in p may be used as the low-rank approximation as long as $\ell \leq L/2$ whereas the interpolant in x is a valid low-rank approximation as soon as $\ell \geq L/2$. Therefore, at $\ell = L/2$, we need to switch representation. Recall that for $\ell \leq L/2$ the expansion coefficients $\{\delta_t^{AB}, 1 \leq t \leq r_\epsilon\}$ may be regarded as equivalent sources while for $\ell \geq L/2$, they approximate the values of the potential $u^B(x)$ on the Chebyshev grid $\{x_t^A, 1 \leq t \leq r_\epsilon\}$. Hence, for any pair (A, B) with A at level $L/2$ (and likewise for B), we have $\delta_t^{AB} \simeq u^B(x_t^A)$ from (12) so that we may set

$$\delta_t^{AB} = \sum_s a(x_t^A, y_s^B) e^{iM\Phi(x_t^A, y_s^B)}\, \delta_s^{AB} \tag{15}$$

(we abuse notations here since $\{\delta_t^{AB}\}$ denotes the new set of coefficients and $\{\delta_s^{AB}\}$ the older set).

5. *Recursion (end).* The rest of the recursion is analogous. For $\ell = L/2 + 1, \ldots, L$, construct the coefficients $\{\delta_t^{AB}, 1 \leq t \leq r_\epsilon\}$ as follows. With $\{\alpha_t^{AB}\}$ and $\{\beta_t^{AB}\}$ given by (11), we have

$$u^B(x) = \sum_c u^{B_c}(x) \simeq \sum_{t',c} \alpha_{t'}^{A_p B_c}(x) \sum_{p \in B_c} \beta_{t'}^{A_p B_c}(y) f(y) \simeq \sum_{t',c} \alpha_{t'}^{A_p B_c}(x) \delta_{t'}^{A_p B_c}.$$

Hence, since δ_t^{AB} should approximate $u^B(x_t^A)$ by (12), we simply set

$$\delta_t^{AB} = \sum_{t',c} \alpha_{t'}^{A_p B_c}(x_t^A) \delta_{t'}^{A_p B_c}.$$

Substituing α_t^{AB} with its value gives the update

$$\delta_t^{AB} = \sum_c e^{iM\Phi(x_t^A, y_0(B_c))} \sum_{t'} \left(L_{t'}^{A_p}(x_t^A) e^{-iM\Phi(x_{t'}^{A_p}, y_0(B_c))} \delta_{t'}^{A_p B_c} \right). \tag{16}$$

6. *Termination.* Finally, we reach $\ell = L$ and set B to be the root box of T_P. For each leaf box A of T_X, we have

$$u^B(x) \simeq \sum_t \alpha_t^{AB}(x) \delta_t^{AB}, \quad x \in A,$$

where $\{\alpha_t^{AB}\}$ is given by (11). Hence, for each $x \in A$, we set

$$u(x) = e^{iM\Phi(x, y_0(B))} \sum_t \left(L_t^A(x) e^{-iM\Phi(x_t^A, y_0(B))} \delta_t^{AB} \right). \tag{17}$$

Most of the computation is in (14) and (16). Because of the tensor product structures, the computations in (14) and (16) can be accelerated by performing Chebyshev interpolation one dimension at a time, reducing the number operations from $O(q^4)$ to $O(q^3)$, where q is the size of the Chebyshev grid in each dimension. As there are at most $O(M^2 \log M)$ pairs of boxes (A, B), the recursion steps take at most $O(q^3 M^2 \log M)$ operations. The cost of (15) is of order $O(q^4 M^2)$ operations since for each pair (A, B) on the middle level a $q^2 \times q^2$ linear transformation is required. Hence, the overall complexity estimate of this algorithm $O(q^3 M^2 \log M + q^4 M^2) = O(q^3 N \log N + q^4 N)$. For brevity we bound this further as $O(q^4 N \log N)$. For the purpose of the rigorous error estimate, q depends on M and $1/\epsilon$ logarithmically.

3. NUMERICAL RESULTS

3.1 Spotlight SAR, the Gotcha dataset

In this section we demonstrate the technique on the Air Force Research Laboratory's publicly-released "Volumetric SAR Data Set, Version 1.0".[11] The imaging operator we applied is as follows,

$$\mathcal{I}(i_1, i_2) = \sum_{(j_1, j_2) \in J} e^{i\,\Phi(i_1, i_2, j_1, j_2)}\, A(i_1, i_2, j_1, j_2)\, D(j_1, j_2), \qquad (i_1, i_2) \in I, \tag{18}$$

where

- I, J are two subsets of $[0,1]^2$,
- D is the phase history data, $D(j_1, j_2) = d(\omega(j_1), \gamma(j_2))$,
- the phase $\Phi(i_1, i_2, j_1, j_2) = \frac{2}{c}\omega(j_1)\left(||\gamma(j_2) - \mathbf{x}(i_1, i_2)|| - r_0(j_2)\right)$, and
- the amplitude $A(i_1, i_2, j_1, j_2) = ||\gamma(j_2) - \mathbf{x}(i_1, i_2)||^2 w_1(j_1) w_2(j_2)$,
- γ the flight path in $\mathbb{R}^3$,
- ω spans the angular frequency range of the LFM chirp (in radians per second),
- r_0 the range to scene center from the antenna location (its presence is only due to some preprocessing that needs to be undone),
- and with w_1, w_2 numerical weights allowing for irregularly sampled data.

The butterfly algorithm needs to evaluate A, D and Φ at arbitrary points (j_1, j_2) in between sample points, so we interpolate the data d, γ, r_0, ω; in these experiments we made use of the cubic spline interpolants of the GNU Scientific Library. The scene is defined by $\{\mathbf{x}(i_1, i_2) : (i_1, i_2) \in [0,1]^2\}$, and may be a general surface in $\mathbb{R}^3$ parameterized by i_1, i_2; here we chose $\mathbf{x}(i_1, i_2) = ((b_1 - a_1)i_1 + a_1, (b_2 - a_2)i_2 + a_2, 0)$, with a_1, b_1, a_2, b_2 defining the boundary of the scene.

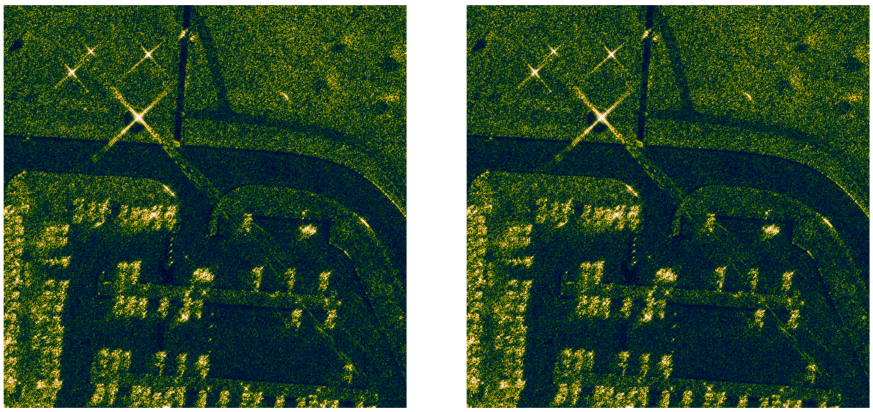

Figure 2. SAR images formed from the Gotcha data set, using the butterfly algorithm with $\sqrt{N} = 2^{10}$, and 4 degrees of azimuthal range, rendered on a logarithmic scale. Left image: $q = 4$, relative root-mean-square (RMS, ℓ_2) error of 3.2e-2, speedup of 402. Right image: $q = 17$, relative RMS error of 1.4e-3, speedup of 3.0. The image on the right is visually indistinguishable from that formed by performing direct summation.

The total number of available frequency samples is $M_\omega = 426$. In the spotlight configuration, the position $\gamma(s)$ of the aircraft is indexed by the "azimuthal range" s. The latter is an angle: it is first partitioned into 360 1-degree angular sectors, and within each sector by 118 angular samples. The images above are obtained by using only 4 contiguous sectors. We let M_γ the total number of angular samples; in the example above, $M_\gamma = 472$.

The current version of the butterfly algorithm handles square domains and boxes. A rectangular data space Y or image space X can easily be accommodated by stretching and resampling the dataset in the direction of least number of samples. This results in a total number of samples given by

$$N = \left(\max\{M_\omega, M_\gamma\}\right)^2.$$

The two tuning parameters are q, the number of Chebyshev points per dimension per square, and L the depth of the tree. The slight oversampling mentioned above — when M_ω, M_γ are larger than M — alleviates the need for considering values of q greater than 10 or 15 in practice. L should be chosen in such a way that each leaf box in the data domain Y contains fewer than q^2 points, and such that the size of the leaf boxes in image space X matches the desired level of resolution.

4. CONCLUSION

The ideas of the butterfly scheme are of course very reminiscent of the fast multipole method (FMM).[18,24] The authors believe that the butterfly algorithm is the proper way to generalize FMM in an all-purpose way to settings where high-frequency oscillations are present. "All-purpose" means robustness, and applicability to many problems, but it also means that other numerical methods may be faster for certain problems with structure. The fast Fourier transform, and the USFFT,[15] are examples of algorithms which are faster than the butterfly, but which only work for bilinear phases. The table below summarizes ballpark complexity figures and ranges of applicability for the FFT, the USFFT of Dutt and Rokhlin, and the Chebyshev butterfly (B-Cheb) algorithm presented here. The figures are for the one-dimensional transforms, so the complexity multiplier should be squared for the two-dimensional transforms.

Kernel	Algorithm	Complexity vs. FFT
e^{ixk}	FFT	1
$e^{ix_jk_n}$	USFFT	6
$a(x,k)e^{i\phi(x,k)}$, $\phi(x,\alpha k) = \alpha\phi(x,k)$	B-Cheb	30

The variable k is supposed to take on large values, on the order of N, in all cases. By x_jk_n, it is meant xk sampled unevenly. The relation $\phi(x,\alpha k) = \alpha\phi(x,k)$ (for $\alpha > 0$) is a homogeneity condition that the butterfly requires, or very nearly so, for operating at the $N \log N$ complexity level. It is ubiquitous in applications to wave propagation.

5. ACKNOWLEDGMENT

LD would like to thank Stefan Kunis for early discussions on error propagation analysis in the butterfly algorithm. MF is grateful for AFOSR support from Arje Nachman. MF and NM were partially supported by the AFRL Automatic Target Recognition Center. The content of this paper was extracted directly from the publicly-released document #88 ABW-10-5343 without modification.

REFERENCES

1. F. Andersson, R. Moses, and F. Natterer, Fast Fourier Methods for Synthetic Aperture Radar Imaging *IEEE Transactions on Aerospace and Electronic Systems, in press.*
2. S. Basu and Y. Bresler $O(N^2 \log N)$ filtered backprojection algorithm for tomography *IEEE Trans. Im. Proc.* **9-10** (2000) 1760–1773
3. G. Beylkin, The inversion problem and applications of the generalized Radon transform, *Comm. Pure Appl. Math.* **37** (1984) 579–599.
4. G. Beylkin, On the fast Fourier transform of functions with singularities, *Appl. Comput. Harmon. Anal.* **2-4** (1995) 363–381
5. J. Boyd, *Chebyshev and Fourier spectral methods* Dover Publications, Mineola, 2001.
6. A. Brandt, Multilevel computations of integral transforms and particle interactions with oscillatory kernels *Comput. Phys. Commun.*, **65** (1991) 2438
7. M. Brandfass, W. C. Chew Microwave imaging as applied to remote sensing making use of a multilevel fast multipole algorithm *Proc. SPIE 4053* (2000) 52–63
8. E. Candes, L. Demanet, D. Donoho, L. Ying, Fast Discrete Curvelet Transforms *SIAM Multiscale Model. Simul.* **5-3** (2006) 861-899
9. E. Candès, L. Demanet, L. Ying, Fast Computation of Fourier Integral Operators *SIAM J. Sci. Comput.* 29:6 (2007) 2464–2493.

10. E. Candès, L. Demanet, L. Ying, A Fast Butterfly Algorithm for the Computation of Fourier Integral Operators *SIAM Multiscale Model. Simul.* 7:4 (2009) 1727–1750
11. C. Casteel, L. Gorham et al, M. Minardi, S. Scarborough, K. Naidu, A challenge problem for 2D/3D imaging of targets from a volumetric data set in an urban environment *Algorithms for Synthetic Aperture Radar Imagery XIV, Proc. SPIE 6568* (2007)
12. M. Cheney and B. Borden *Fundamentals of radar imaging*, CBMS regional conf. series in applied mathematics, SIAM, Philadelphia, 2009.
13. L. Demanet and L. Ying Fast wave computation via Fourier integral operators *Submitted*, 2010
14. Y Ding, DC Munson A fast back-projection algorithm for bistatic SAR imaging *Proc. IEEE Int. Conf. on Image Processing* **2** (2002) 441–444
15. A. Dutt and V. Rokhlin Fast Fourier Transforms for Nonequispaced Data, II *Applied and Computational Harmonic Analysis* **2-1** (1995) 85–100
16. B. Engquist and L. Ying. Fast directional multilevel algorithms for oscillatory kernels. *SIAM Journal on Scientific Computing* **29-4** (2007) 1710–1737
17. L. Fox and I. B. Parker, *Chebyshev polynomials in numerical analysis* Oxford University Press, Oxford, UK, 1968.
18. L. Greengard, V. Rokhlin, A fast algorithm for particle simulations, *Journal of Computers and Physics* **73** (1987) 325–348.
19. M. Gu, S. Eisenstat, Efficient algorithms for computing a strong rank-revealing QR factorization, *SIAM J Sci Comput* **17-4** (1996) 848–869.
20. P.-G. Martinsson and V. Rokhlin. A fast direct solver for scattering problems involving elongated structures. *J. Comput. Phys.* **221-1** (2007) 288302.
21. E. Michielssen and A. Boag, A multilevel matrix decomposition algorithm for analyzing scattering from large structures *IEEE Transactions on Antennas and Propagation* **44** (1996) 1086–1093
22. S. Nilsson, L. Andersson Application of fast backprojection techniques for some inverse problems of synthetic aperture radar *Proc. SPIE* **3370** (1998) 62–72
23. M. O'Neil and V. Rokhlin, A new class of analysis-based fast transforms *Tech. Rep. 1384, Department of Computer Science, Yale University, August 2007.*
24. V. Rokhlin Rapid solution of integral equations of scattering theory in two dimensions *J. Comput. Phys.* **86-2** (1990) 414–439
25. W. Rudin, *Real and Complex analysis*, 3rd ed. McGraw-Hill ed., Singapore, 1987.
26. M. Soumekh, *Synthetic Aperture Radar Signal Processing with Matlab Algorithms* J. Wiley & Sons (Interscience Div.), 1999.
27. E. Tadmor, The exponential accuracy of Fourier and Chebyshev differencing methods *SIAM J. Num. Analysis*, 23:1 (1986) 1–10
28. N. Trefethen, *Spectral methods in Matlab* SIAM ed., Philadelphia, 2000.
29. M. Tygert Fast algorithms for spherical harmonic expansions, III *Preprint*, 2010.
30. L. Ulander, H. Hellsten, and G. Stenstrom, Synthetic-aperture radar processing using fast factorized back-projection *IEEE Trans. Aero. Elect. Systems* **39-3** (2003) 760–776
31. D. Wahl, P. Eichel, D. Ghiglia, P. Thompson, C. Jakowatz *Spotlight-Mode Synthetic Aperture Radar: A Signal Processing Approach* Springer, 1996.
32. A. Yegulalp Fast backprojection algorithm for synthetic aperture radar *Proc. IEEE radar conf.* (1999) 60–65
33. L. Ying Sparse Fourier transform via butterfly algorithm. *SIAM Journal on Scientific Computing*, **31** (2009) 1678
34. L. Ying, G. Biros, D. Zorin, A kernel-independent adaptive fast multipole method in two and three dimensions *Journal of Computational Physics*, **196-2** (2004) 591–626
35. S. Xiao, D.C. Munson, S. Basu, Y. Bresler An $N^2 \log N$ back-projection algorithm for SAR image formation *Proc. 34th Asilomar Conf. on Signals, Systems and Computers* Pacific Grove, CA, Nov. 2000, pp. 37.