2-D PHASE RETRIEVAL BY PARTITIONING INTO COUPLED 1-D PROBLEMS USING DISCRETE RADON TRANSFORMS

Hyunduk Ahn

Dept. of EECS, The University of Michigan, Ann Arbor, MI 48109-2122 aey@eecs.umich.edu

ABSTRACT

The discrete phase retrieval problem is to reconstruct a discrete-time signal whose support is known and compact from the magnitude of its discrete Fourier transform. We solve the 2-D discrete phase retrieval problem by partitioning it into a mostly-DECOUPLED set of 1-D phase retrieval problems. The discrete and modulated Radon transforms are used to formulate two coupled 1-D problems, the solution to which then specifies solutions to the other DECOUPLED 1-D problems. The latter may in turn be solved in parallel; however, using the solution to one problem as the input to a neighboring problem reduces the computation significantly for serial computers. Unlike other exact 2-D phase retrieval methods which rely on tracking zero curves of algebraic functions or equivalent operations, no continuous-function-based methods are used here. This makes the procedure more robust numerically.

1. INTRODUCTION

The problem of reconstructing a signal (1-D) or image (2-D) known to have compact support from its Fourier transform magnitudes arises in several disciplines [1]. The signal or image is reconstructed if the missing Fourier phase is recovered; hence the term "phase retrieval." For details on the history and applications of this problem see [1]. Since the signal or image has compact support, its Fourier transform may be sampled in frequency. Also, the signal or image is often assumed to be bandlimited, so that it can be sampled in the time domain as well. This leads to the discrete version of this problem, in which a discrete-time signal known to have compact support is to be reconstructed from the magnitude of its discrete Fourier transform (DFT). For details on discrete phase retrieval problems see [2]-[3].

The most common approach for phase retrieval problems is to use one of the iterative transform algorithms [1], which alternate between the time and frequency domains. However, these algorithms usually stagnate, failing to converge to a solution. Other methods have been proposed; see references in [1].

In this paper we then solve the 2-D phase retrieval problem by partitioning the 2-D problem into 1-D problems, which can then be solved in parallel. The appropriate solution to each of these multiple-solution 1-D problems is selected by first solving two coupled 1-D problems, obtained using the discrete Radon transform and what we define as the modulated discrete Radon transform. This is quite different from partitioning procedures proposed elsewhere [4].

Our method is exact and is guaranteed to obtain the solution in a finite number of operations, in contrast to iterative transform methods which almost always stagnate on 2-D problems. It also does not require the extremely unstable numerical operation of tracking zero curves of algebraic functions, which has been proposed for continuous phase retrieval problems [5] and discrete problems [6]-[7]. Since zero locations vary widely with small changes in polynomial coefficients, these methods are impractical due to numerical roundoff error. Our method is also sensitive to noise, but less so than these other methods, since more than two 1-D problems can be coupled together. This produces a redundancy which can handle small amounts of noise. We discuss this elsewhere.

This work was supported by the Office of Naval Research under Young Investigator grant #N00014-90-J-1897

2. FORMULATIONS OF DISCRETE PHASE RETRIEVAL PROBLEMS

2.1. Quick Review of 1-D Discrete Phase Retrieval

The 1-D discrete phase retrieval problem is as follows [2]-[3]. Let $x(n)$ be a discrete-time periodic signal of period N and $X(k)$ be its N-point DFT

$$X(k) = \sum_{i=0}^{N-1} x(i)e^{-j2\pi ik/N}, k = 0, 1, \ldots N-1. \quad (1)$$

Given knowledge that only M consecutive values of $x(n)$ differ from zero (i.e., $x(n)$ has M-point support) and given the values of the DFT magnitudes $|X(k)|$ for $0 \le k \le N-1$, determine $x(n)$ (or equivalently $X(k)$).

There are some trivial ambiguities in this problem. Clearly if $x(n)$ is a solution then $x^*(-n)$, $cx(n)$ and $x(n-b)$ are also solutions for any integer b and any complex number c having unity magnitude $|c| = 1$ (note the third is actually a special case of the second). We refer to c as the *arbitrary phase factor*. If $x(n)$ is also constrained to be real, then $c = \pm 1$ only. We call these trivial ambiguities the associated solutions to a given solution $x(n)$.

Excluding these associated solutions, there are almost surely 2^M solutions to the discrete 1-D phase retrieval problem. This can be seen by noting that the set of squares of the DFT magnitudes is the DFT of the autocorrelation $r(n) = x(n) * x^*(-n)$ of $x(n)$. The z-transform of $r(n)$ is $X(z)X^*(1/z^*)$, where $X(z) = \sum_{n=0}^{M-1} x(n)z^n$ is the z-transform of $x(n)$ (without loss of generality the M-point support of $x(n)$ is specified as interval $[0, M-1]$). Since

 0-7803-2431-5/95 $4.00 © 1995 IEEE

$X(z)$ is a polynomial of degree $M-1$, $z^{*(M-1)}X(z)X^*(1/z^*)$ is a polynomial of degree $2M-2$ whose zeros occur in conjugate reciprocal pairs (if z is a zero then $1/z^*$ is also a zero). One of each of these reciprocal pairs of zeros must be chosen to form $X(z)$ so that $x(n)$ will have M-point support; this can be done in 2^{M-1} different ways (if any zero lies on the unit circle $z = 1/z^*$, this choice disappears for that zero). If the associated solution $x^*(-n)$ is counted as a solution distinct from $x(n)$, the number of solutions increases to 2^M. We consider these to be distinct solutions later.

If $x(n)$ is real, the zeros must be chosen in complex conjugate pairs, so there are only $2^{(M-1)/2}$ solutions if all of the zeros of $X(z)$ are complex (real zeros have no complex conjugate requirement; note that if M is even at least one zero is real). For a signal with randomly chosen values $x(n)$, then almost surely there are no zeros on the unit circle and no more than one real zero, so we can use the above expressions for determining computational complexity. Here "almost surely" means that the set of exceptions has Lebesgue measure zero.

2.2. 2-D Discrete Phase Retrieval

The discrete 2-D phase retrieval problem is simply the 2-D version of the 1-D problem defined above [2]-[3]. It almost surely has a unique solution (to the trivial ambiguities listed above), for images whose values have been selected at random. In the sequel, we assume that the given 2-D Fourier magnitude data are such that there is exists a unique solution to the phase retrieval problem.

3. DISCRETE AND MODULATED RADON TRANSFORMS

Recall that the $(N \times N)$-point 2-D DFT is defined as

$$X(k_1,k_2) = \sum_{i_1=0}^{N-1}\sum_{i_2=0}^{N-1} x(i_1,i_2)e^{-j2\pi(i_1k_1+i_2k_2)/N}. \quad (2)$$

Let $x(i_1,i_2)$ have $(M \times M)$ support, where $N \geq 2M$ to avoid aliasing. Setting $k_1 = k_2 = k$ in (2) and recognizing the 2N-point 1-D DFT (1) gives $X(k,k) =$

$$\sum_{i_1=0}^{N-1}\sum_{i_2=0}^{N-1} x(i_1,i_2)e^{-j2\pi(i_1+i_2)k/N} = \sum_{i=0}^{2N-1} \tilde{x}(i)e^{-j2\pi i2k/(2N)} \quad (3)$$

where

$$\tilde{x}(i) = \sum_{i_1=0}^{M-1}\sum_{i_2=0}^{M-1} x(i_1,i_2)\delta(i-i_1-i_2), 0 \leq i \leq (2M-1) \quad (4)$$

is the *discrete Radon transform* of $x(i_1,i_2)$ at 45 degrees.

Note that $\tilde{x}(i)$ is the sum of all values of $x(i_1,i_2)$ along lines of slope -1 (at an angle of 135 degrees). That this is also the 2N-point inverse 1-D DFT of $X(k_1,k_2)$ along the slice $k_1 = k_2$ is the discrete projection slice theorem of the discrete Radon transform. Note that a 2N-point 1-D DFT is required in (3) to avoid aliasing, since $\tilde{x}(i)$ has $2M-1$-point support. Also note that the 2N-point 1-D DFT of $\tilde{x}(i)$ is zero for odd values of frequency, but only even values of frequency are needed for $X(k,k)$ (see (3)). This can be understood by noting that the separation between constant lines of i_1+i_2 is $1/\sqrt{2}$, not 1, and the spacing between successive values of $X(k,k)$ is $\sqrt{2}$, agreeing with the scaling property of Fourier transforms.

Now let N be even, and set $k_1+k_2 = N/2$ in (2):

$$X(k_1,k_2 = N/2-k_1) = \sum_{i_1=0}^{N-1}\sum_{i_2=0}^{N-1} x(i_1,i_2)e^{-j2\pi(i_1k_1+i_2(N/2-k_1))/N}$$

$$= \sum_{i_1=0}^{N-1}\sum_{i_2=0}^{N-1} x(i_1,i_2)(-1)^{i_2}e^{-j2\pi(i_1-i_2)k_1/N} = \sum_{i=0}^{N-1} \hat{x}(i)e^{-j2\pi i2k_1/(2N)} \quad (5)$$

where we now define

$$\hat{x}(i) = \sum_{i_1=0}^{M-1}\sum_{i_2=0}^{M-1} x(i_1,i_2)(-1)^{i_2}\delta(i-i_1+i_2), -(M-1) \leq i \leq (M-1) \quad (6)$$

as the *modulated* (by $(-1)^{i_2}$) discrete Radon transform of $x(i_1,i_2)$ at 135 degrees. This transform seems to be new.

Note that $x(i_1,i_2)$ is multiplied by $(-1)^{i_2}$, but this obviously does not affect the support or real-valuedness of $x(i_1,i_2)$. $\hat{x}(i)$ is the sum of all values of $x(i_1,i_2)(-1)^{i_2}$ along lines of slope 1 (at an angle of 45 degrees). Again a 2N-point 1-D DFT is required in (5) to avoid aliasing, since $\hat{x}(i)$ varies from $-(M-1)$ to $M-1$.

4. PARTITIONING ALGORITHM

4.1. Partitioning

Our partitioning algorithm is based on the fact (which we have shown elsewhere) that a 1-D phase retrieval problem with a single specified phase value (after fixing the arbitrary phase factor c) almost surely has a unique solution. That is, if $X(k)$, not just $|X(k)|$, is known for some single $k \neq 0, N/2$, then this picks out one of the 2^M solutions of the problem. This is a Fourier domain analogue of the well-known result that a 1-D phase retrieval problem with a specified endpoint $x(0)$ almost surely has a unique solution [8]-[9].

Indeed, the specified phase value result is essentially equivalent to the specified endpoint problem. To see this, suppose that there are two solutions $X_1(k)$ and $X_2(k)$ to a 1-D phase retrieval problem that have the same phase at $k = k_0$. This means that the z-transforms $X_1(z)$ and $X_2(z)$ of the two solutions $x_1(n)$ and $x_2(n)$ agree at $z = e^{-j2\pi k_0/N}$. Without loss of generality, we can scale z by a constant ρ; this multiplies the $x_i(n)$ and their (identical) autocorrelations $r(n)$ by ρ^n, but there are still two different solutions whose z-transforms agree at $z = \rho e^{-j2\pi k_0/N}$. Letting $\rho \to 0$, $X_i(\rho e^{-j2\pi k_0/N}) \to X_i(0) = x_i(0)$, so that we have two different solutions having the same endpoint $x_1(0) = x_2(0)$. Hence a nonunique solution specified endpoint problem can be associated with each nonunique solution specified phase problem. Thus the specified phase value problem, like the specified endpoint problem, almost surely has a unique solution.

4.2. New Algorithm

The $M \times M$ support 2-D phase retrieval problem can be solved as follows:

1. Solve the two coupled 1-D phase retrieval problems of reconstructing $\tilde{x}(i)$ (defined in (4)) and $\hat{x}(i)$ (defined in (6)) from their given DFT magnitudes $|X(k,k)|$ and $|X(k,N/2-k)|$ (see (3) and (5)). Since $\tilde{x}(i)$ and $\hat{x}(i)$ are both real, there is no arbitrary phase factor c in either problem, and the phases $X(N/4,N/4)$ must agree. This picks out one of the 2^{2M} solutions for each problem, and specifies the phases $X(k,k)$ and $X(k,N/2-k)$ for all k.
2. Solve the M *decoupled* 1-D phase retrieval problems of reconstructing the phase of each row of $X(k_1,k_2)$ from $|X(k_1,k_2)|$. That is, fix k_1 and solve the resulting 1-D phase retrieval problem in k_2, and do this for each k_1. Note that since $x(n_1,n_2)$ has $M \times M$ support, its half-2-D transform $X(k_1,n_2)$ still has M-point support in n_2. Each problem has its phase specified at two points $X(k_1,k_1)$ and $X(k_1,N/2-k_1)$, so the arbitrary phase factor c is determined and a single solution picked out. Once all of these problems have been solved, the 2-D phase has been recovered everywhere, so the entire problem has been solved (the single row $k_1 = N/4$ may be handled separately).
3. The specified-phase 1-D problems can be solved using any of the methods listed below.

4.3. Alternatives for Solving 1-D Problems

We have devised several methods for solving the decoupled 1-D problems. The optimal method depends on the computing environment (is serial or parallel computation available?):

1. The most straigntforward method consists of finding the zeros of the z-transform of the autocorrelation, computing the phase contribution from each zero at the point $X(k_0)$ of specified phase, and determining the phase of $X(k_0)$ for each of the 2^M choices between each of the M zeros and their reciprocal conjugates by simply adding the M phase contributions. Each choice can thus be checked with just a few additions. This method is ideal if parallel or distributed computation using only very simple processors is available, since the 1-D problems are all completely decoupled and the operations required are trivial (additions and checking equalities).
2. A branch-and-bound algorithm offers significant improvement over an exhaustive search; such an algorithm was successfully used for the specified endpoint phase retrieval problem [9]. We have developed a similar algorithm for the present problem. This can also be done using parallel or distributed computation, although the operations are now more complex.
3. We have discovered that the zero configuration (viz., inside or outside the unit circle) does not change much from one 1-D problem to its neighbor. This allows the solution to a 1-D problem to furnish an excellent initialization to its neighbor. Proceeding in sequence, we can solve all of the 1-D problems in succession. We have found that only a minor search is required every so often, depending on how many zeros have flipped from inside to outside the unit circle (or vice versa). This is the fastest algorithm on a serial machine. However, it cannot be parallelized, since the 1-D problems are solved in sequence.
4. Iterative transform algorithms [1] can be used for the 1-D problems. We have been unable to discover a clear preference here–sometimes they work better than the above methods, sometimes they do not converge at all. This unpredictability makes them unsuitable, in our view.

5. NUMERICAL RESULTS

We compare our method to the hybrid I/O algorithm [1] on the 19×19 image ($M = 19, N = 64$) shown in Fig. (a). Our method, implemented with a branch and bound strategy, reconstructed the image perfectly after checking 40012893 leaves. The hybrid I/O method was initialized with three random-phase images. Twice it converged and once, when initiated with Fig. (b), it stagnated at Fig. (d). Its performance vs. #iterations is shown in Fig. (c).

6. REFERENCES

[1] J.C. Dainty and J.R. Fienup, "Phase Retrieval and Image Reconstruction for Astronomy," in *Image Recovery: Theory and Application*, ed. by H. Stark, Academic Press, San Diego, 1987, pp. 231-275.

[2] M.H. Hayes, "The Reconstruction of a Multidimensional Sequence from the Phase or Magnitude of its Fourier Transform," IEEE Trans. ASSP 30(2), 140-154, April 1982.

[3] M.H. Hayes, J.S. Lim, and A.V. Oppenheim, "Signal Reconstruction from Phase or Magnitude," IEEE Trans. ASSP 28(6), 672-680, Dec. 1980.

[4] H.V. Deighton, M.S. Scivier and M.A. Fiddy, "Solution of the Two-Dimensional Phase Retrieval Problem," Opt. Lett. 10, 250-251, 1985.

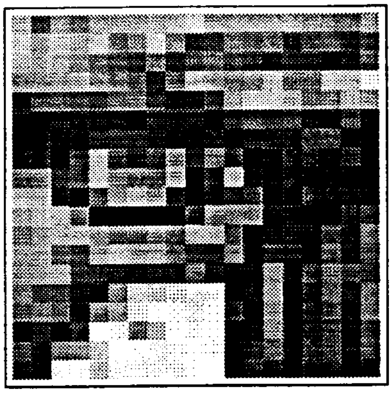
[5] P.J. Napier and R.H.T. Bates, "Inferring Phase Information from Modulus Information in Two-Dimensional Aperture Synthesis," Astron. Astrophys. Suppl. 15, 427-430, 1974.

[6] D. Izraelevitz and J.S. Lim, "A New Direct Algorithm for Image Reconstruction from Fourier Transform Magnitude," IEEE Trans. ASSP 35, 511-519, April 1987.

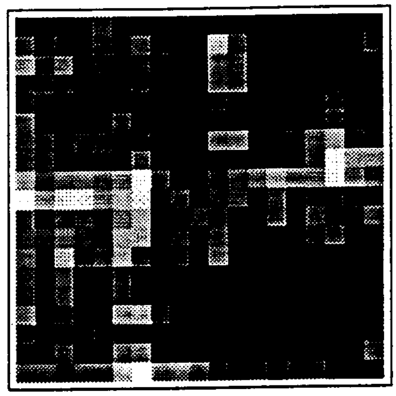
[7] R.G. Lane, W.R. Fright and R.H.T. Bates, "Direct Phase Retrieval," ibid., 520-526.

[8] L. Xu, P. Yan, and T. Chang, "Phase Information is Richly Contained in the End Point of a Finite Length Digital Signal," Proc. IEEE Int. Symp. on Circuits and Systems, vol. 3, 851-854, 1987.

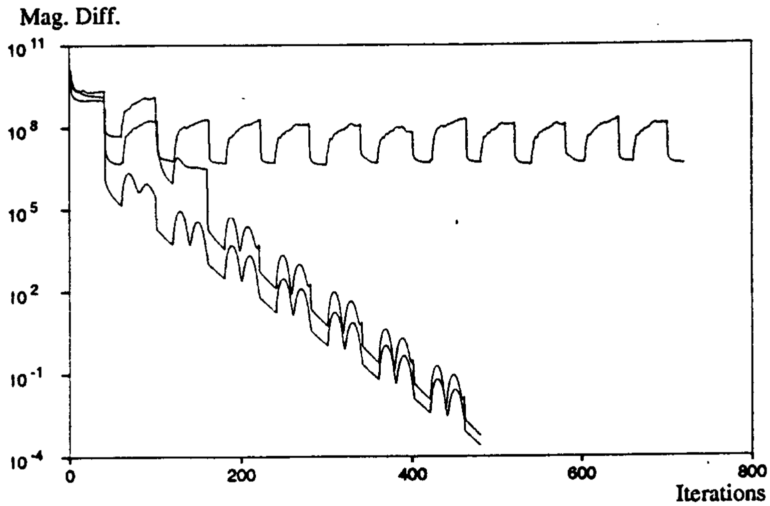
[9] H. Sahinoglou and S. Cabrera, "On Phase Retrieval of Finite-Length Sequences Using the Initial Time Sample," IEEE Trans. Ckt. and Sys. 38(8), 954-958, 1978.



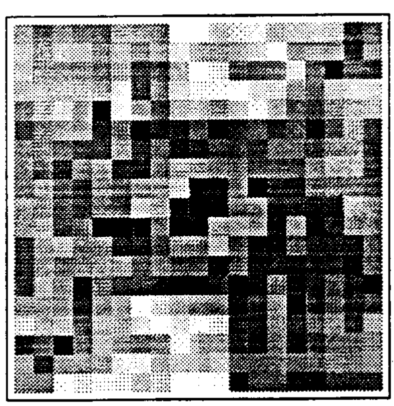
(a)



(b)



(c)



(d)

Comparison of the paritioning algorithm and the hybrid I/O algorithm. (a) the image used; (b) an initial estimate; (c) performances of three runs of the hybrid I/O algorithm; (d) stagnated image.