Improved ultrashort pulse-retrieval algorithm for frequency-resolved optical gating

Kenneth W. DeLong and Rick Trebino

Combustion Research Facility, Sandia National Laboratories, Livermore, California 94551

Received December 10, 1993; revised manuscript received March 21, 1994; accepted March 22, 1994

We report on significant improvements in the pulse-retrieval algorithm used to reconstruct the amplitude and the phase of ultrashort optical pulses from the experimental frequency-resolved optical gating trace data in the polarization-gate geometry. These improvements involve the use of an intensity constraint, an overcorrection technique, and a multidimensional minimization scheme. While the previously published, basic algorithm converged for most common ultrashort pulses, it failed to retrieve pulses with significant intensity substructure. The improved composite algorithm successfully converges for such pulses. It can now retrieve essentially all pulses of practical interest. We present examples of complex waveforms that were retrieved by the improved algorithm.

1. INTRODUCTION

The technique of frequency-resolved optical gating (FROG) has proven to be quite useful for the measurement of the amplitude and the phase of an arbitrary ultrashort optical pulse.[1,2] Experimentally, the technique involves the generation of a spectrogram,[3] known as the FROG trace, of the ultrashort pulse by means of a simple apparatus. The FROG trace itself is an extremely informative, useful, and visually intuitive representation of the pulse,[4] and it is generally possible to understand the qualitative features of a pulse by simple visual inspection of the trace. In order to determine the quantitative amplitude and phase of the pulse as a function of time, however, one must use a pulse-retrieval algorithm, in which the FROG trace serves as input data.[1]

This basic FROG algorithm[1] has proven to be fairly robust and powerful. It has sufficed in recovering the amplitude and the phase of many experimental and numerically modeled pulses.[1,2,5] However, there is one class of pulses that has proven extremely difficult for the basic FROG algorithm. This class is typified by what we call the double pulse, a waveform with two distinct peaks of roughly equal intensities. The basic FROG algorithm has proven unequal to the challenge of recovering the double pulse from its FROG trace data. In addition, other pulses with complex intensity profiles—for example, sequences of independent pulses—have also defeated this algorithm. While previously these types of pulses have been undesirable, they are becoming important in experiments, especially in pulse-shaping applications. It is therefore important that the FROG technique be effective with this type of pulse.

In this paper we describe improvements to the basic FROG algorithm that enable it to converge to a much larger class of pulses, including the troublesome double pulse, as well as pulses with more complicated intensity and phase profiles. These improvements involve the use of what we have called an intensity constraint, an overcorrection method, and a multidimensional minimization technique. The intensity constraint employs a technique that permits us to determine the gate function used to make the FROG trace, which is identical to the pulse intensity profile. The overcorrection technique overcorrects our guess for the complex signal field where it deviates from the actual FROG trace. The multidimensional minimization scheme treats the sample points in the electric-field envelope of the pulse as independent variables and tries to minimize an error function by adjusting these values. The overcorrection method speeds convergence of otherwise slowly converging pulses, while the intensity-constraint method improves on the results of the basic FROG algorithm and generates a pulse that is close enough to the correct answer for convergence of the minimization routine.

We have also developed a composite algorithm that combines all the above techniques in an intelligent and self-regulating way. The composite algorithm monitors its own progress and combines the above techniques in a way that is heuristically found to be effective.

In constructing pulses to test our algorithm, we have used a certain class of pulse that is physically relevant and representative of pulses that one may expect to encounter in real experiments. Our pulses were constructed as the coherent sum of up to four Gaussian intensity profile pulses of varying heights and widths. Each pulse could be assigned an arbitrary amount of linear chirp or self-phase modulation. We also tested pulses with discrete phase jumps, cubic phase distortion, and other more complicated phase profiles. Using the composite algorithm described here, we can retrieve essentially all of these pulses from the FROG trace. We say "essentially all" to reflect the fact that, although we have not found a pulse in this class that has defeated the algorithm, it is impossible to test all the infinitely many possible pulse shapes.

In this paper we first describe the basic FROG algorithm and detail the stagnation problems associated with pulses with significant intensity substructure, specifically the double pulse. We then detail the improvements that we have made to the basic FROG algorithm. Next we describe the multidimensional minimization schemes used

0740-3232/94/092429-09$06.00

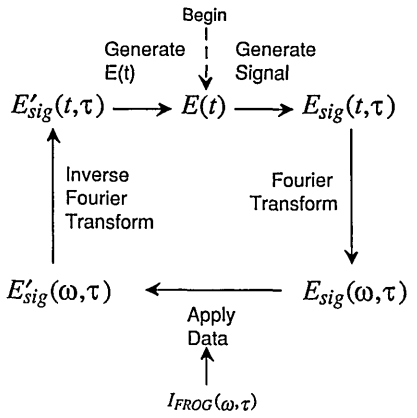

Fig. 1. Schematic of the basic FROG algorithm. The equations used, starting clockwise from $E(t)$, are Eq. (1) (Generate Signal), Eq. (3) (Apply Data), and Eq. (4) [Generate $E(t)$].

in the new composite algorithm. Finally, we show several examples of complex pulses and the varying degrees of success of the old and new algorithms. We also review several techniques that were not effective in improving the basic FROG algorithm.

2. BASIC FROG ALGORITHM

First, we describe the main points of the basic FROG algorithm. The task of the algorithm is to retrieve the complex electric-field envelope $E(t)$ of the pulse given only the FROG trace obtained with the pulse. The algorithm, diagrammed in Fig. 1, is based on the iterative Fourier transform algorithm common in phase retrieval.[6] In a phase-retrieval problem one typically wishes to reconstruct an image from two pieces of information: the intensity of the Fourier transform of the image and an image domain constraint (*a priori* knowledge concerning the image). A common constraint is the knowledge that the image has zero intensity outside a certain bounding area. The common iterative Fourier transform algorithm involves Fourier transforming a trial image back and forth between the image domain and the Fourier domain. The *a priori* constraint is applied in the image domain, and in the Fourier domain the trial image is forced to have the same intensity as that of the known (usually measured) data.

In the basic FROG algorithm there is no physically relevant image domain and thus no *a priori* constraints that we can apply. However, we do have *a priori* knowledge of the nonlinear optical process used to generate the experimental FROG trace, and from this knowledge we know the form of the signal field:

$$E_{\mathrm{sig}}(t,\tau) = E(t)|E(t-\tau)|^2 . \tag{1}$$

[Equation (1) is valid for the polarization-gate geometry. The other common FROG geometries, self-diffraction and second-harmonic generation, will not be treated in this paper. The details of the implementation of the algorithm will differ in the differing geometries, although the general procedure remains the same.] This signal field is Fourier transformed (by a spectrometer in the experiment), and the resultant intensity generates the FROG trace:

$$I_{\mathrm{FROG}}(\omega,\tau) = \left|\int_{-\infty}^{\infty} \mathrm{d}t\, E_{\mathrm{sig}}(t,\tau)\exp(i\omega t)\right|^2 = |E_{\mathrm{sig}}(\omega,\tau)|^2 . \tag{2}$$

It turns out that knowledge of these equations and the proper application of that knowledge are more powerful than the standard phase-retrieval image domain constraints (which, incidentally, have not proven very effective at complex phase retrieval, of which FROG is a special case[7,8]).

The basic FROG algorithm begins with a guess for the complex electric field $E(t)$ (center top of Fig. 1). In standard phase retrieval, noise is used,[6] although we use a smoothly varying intensity (typically Gaussian) with a random phase distribution. One cycle of this iterative algorithm returns a new guess for $E(t)$, one that we hope is closer to the correct answer. A single cycle of the algorithm proceeds as follows.

We use the current guess for $E(t)$ to generate the signal field $E_{\mathrm{sig}}(t,\tau)$ from Eq. (1). We then Fourier transform this signal field [as in Eq. (2)] with respect to time to obtain $E_{\mathrm{sig}}(\omega,\tau)$. The magnitude of this result is then replaced with the magnitude of the experimentally measured FROG trace through

$$E'_{\mathrm{sig}}(\omega,\tau) = \frac{E_{\mathrm{sig}}(\omega,\tau)}{|E_{\mathrm{sig}}(\omega,\tau)|}\,[I_{\mathrm{FROG}}(\omega,\tau)]^{1/2} , \tag{3}$$

just as in the usual phase-retrieval problem. The phase of $E_{\mathrm{sig}}(\omega,\tau)$ is left unchanged. We then inverse Fourier transform this new signal field, $E'_{\mathrm{sig}}(\omega,\tau)$, to yield $E'_{\mathrm{sig}}(t,\tau)$. In order to complete the cycle, we generate a new version of $E(t)$ by integrating the signal field $E'_{\mathrm{sig}}(t,\tau)$ with respect to τ and use this result as the new guess for $E(t)$. If it is performed on the correct signal field, this integration becomes, according to Eq. (1),

$$\begin{aligned}\int_{-\infty}^{\infty} \mathrm{d}\tau\, E_{\mathrm{sig}}(t,\tau) &= \int_{-\infty}^{\infty} \mathrm{d}\tau\, E(t)|E(t-\tau)|^2 \\ &= E(t)\int_{-\infty}^{\infty} \mathrm{d}\tau |E(t-\tau)|^2 \\ &= CE(t) ,\end{aligned} \tag{4}$$

where C is a real-valued constant proportional to the pulse energy.

The inverse Fourier transform that takes $E'_{\mathrm{sig}}(\omega,\tau)$ to $E'_{\mathrm{sig}}(t,\tau)$ and the subsequent integration with respect to τ are interchangeable. Therefore, in practice, we first integrate $E'_{\mathrm{sig}}(\omega,\tau)$ with respect to τ and then inverse Fourier transform to obtain the new $E(t)$. In this way we reduce the number of inverse Fourier transform operations from N to 1, thus increasing the speed of the algorithm.

We then begin the cycle anew, using this new field. If the current guess for $E(t)$ is the correct one (which generates the proper FROG trace), the replacement step of Eq. (3) leaves $E_{\mathrm{sig}}(\omega,\tau)$ unchanged. This means that the new $E(t)$ generated by Eq. (4) will be the same as the field that began the cycle, so that a correct pulse remains unchanged through a cycle of this algorithm. An incorrect pulse will usually approach the correct one, although nothing presented here guarantees it. (Occasionally, there are also incorrect pulses that remain un-

changed through a single cycle of the algorithm. These pathological cases are more fully discussed in Section 4.)

The basic FROG algorithm iterates, alternately applying the experimentally measured FROG trace data and enforcing compliance with the form of the signal field in Eq. (1). After several tens of iterations (typically 20–70) the algorithm has usually converged. On a typical-size FROG trace of 64 × 64 pixels each iteration of the basic FROG algorithm takes less than 1 s on a computer based on a 50-MHz Intel 80486, so that the whole process of retrieving a pulse from its FROG trace data typically takes less than 1 min.

The size of the FROG trace grid imposes a compact support constraint on $E(t)$. Numerically, the electric field is stored in an array of N elements. One can use this field to generate an $N \times N$ FROG trace. In this case the field should have negligible intensity outside a width of $N/2$, or else nonnegligible portions of the FROG trace will lie outside the numerical $N \times N$ FROG trace grid. If this becomes a problem, one can increase the size of the FROG trace grid, with a concurrent increase in computational complexity.

3. MEASURING ERROR

In order to assess the progress of the algorithm, we need a measure describing how closely the algorithm has converged, i.e., how close the current guess for $E(t)$ is to the field that generated the FROG trace data. In practice, we cannot measure this quantity directly, since we do not know the original field (that is what we are looking for!).

In a previous publication[1] we showed that there is a one-to-one correspondence between all possible fields and their FROG traces. This means that if two FROG traces are identical, then the fields that generated them are identical; whereas, if the FROG traces differ, then the fields also must differ. Thus a meaningful measure of convergence is the difference between the experimental FROG trace $I_{\mathrm{FROG}}(\omega, \tau)$ and the FROG trace generated by the current guess for $E(t)$. This difference will serve as an error that measures how closely the FROG trace has retrieved the correct field.

The difference between the two FROG traces is quantified by the error

$$G = \left\{ \frac{1}{N^2} \sum_{\omega,\tau=1}^{N} [I_{\mathrm{FROG}}(\omega, \tau) - |E_{\mathrm{sig}}(\omega, \tau)|^2]^2 \right\}^{1/2}, \quad (5)$$

which is the root-mean-square error of the difference between the experimentally generated FROG trace I_{FROG} and the squared magnitude of the signal field generated by the current guess for $E(t)$. Before the error is calculated, both are normalized to a peak intensity of unity. Here ω and τ index the elements of the two-dimensional arrays that hold I_{FROG} and E_{sig}, and they take on integer values.

In order to get an idea of how sensitive G is to variations in the pulse, we have computed the value of G for various pairs of fields that are similar to each other but differ in some small way. The calculations were made on fields sampled at 64 points, leading to a 64 × 64 pixel FROG trace. For this set of examples we used a pulse electric field that is given by the sum of two Gaussian pulses:

$$E(t) = E_1(t) + E_2(t), \quad (6)$$

with

$$E_1(t) = \exp\{-2(\ln 2)(t/t_p)^2 + i[At^2 + Q|E_1(t)|^2]\}, \quad (7)$$

$$E_2(t) = B \exp\{-2(\ln 2)[(t - D)/t_p]^2 + i[A(t - D)^2 + Q|E_2(t)|^2]\}, \quad (8)$$

where t_p is the intensity full width at half-maximum (FWHM) of the pulse, A and Q describe the amount of linear chirp and self-phase modulation, respectively, and B is the amplitude and D is the separation of the second pulse in the double-pulse case. The results are shown in Table 1. We see that phase differences of even much less than 1 rad across the FWHM of the pulse result in errors of the order $G = 0.01$. In general, a value of $G = 0.0001$ or less indicates quite good convergence.

4. DOUBLE PULSE

Pulses with significant intensity substructure can present problems for the basic FROG algorithm. Even the case of an electric-field intensity profile with just two distinct peaks, the double pulse, has proven quite difficult. If the intensity of the smaller peak in a double pulse is less than half that of the main peak, the algorithm converges without problem, even in the presence of phase distortions.[1] Indeed, even in pulses with a complicated intensity profile, if the heights of the substructure peaks are less than half that of the main peak, the basic FROG algorithm suffices. For peaks of more equal intensity, however, the basic FROG algorithm fails to converge and instead stagnates at an incorrect solution for $E(t)$. Consequently, we have used the equal-height double pulse as a benchmark pulse for testing improvements to the FROG algorithm.

In Fig. 2 we show the performance of the basic FROG algorithm on two initial pulses, one a pulse with self-phase modulation and the other a double-peaked pulse in which the peaks are of equal intensity and separated by twice their intensity FWHM. From Fig. 2 we see that, although the algorithm easily converges to the self-phase-modulated pulse, it does not converge to the double pulse.

Table 1. Variation of the Error G for Various Pairs of Fields[a]

Field 1	Field 2	G
Single Pulse ($B = 0$)		
$t_p = 8, A = Q = 0$	$t_p = 8, A = 0, Q = 1$	0.0185
$t_p = 8, A = Q = 0$	$t_p = 8, A = 0, Q = 2$	0.0414
$t_p = 8, A = Q = 0$	$t_p = 8, A = 0.01, Q = 0$	0.00988
$t_p = 8, A = Q = 0$	$t_p = 8, A = 0.02, Q = 0$	0.0196
$t_p = 8, A = Q = 0$	$t_p = 7, A = 0, Q = 0$	0.0121
Double Pulses ($t_p = 6, A = Q = 0$)		
$B = 1, D = 12$	$B = 0.99, D = 12$	0.000318
$B = 1, D = 12$	$B = 0.9, D = 12$	0.00352
$B = 1, D = 12$	$B = 1, D = 11$	0.0139

[a]The FROG trace generated by field 1 is compared with that generated by field 2. The error between these two FROG traces is quantified by Eq. (5). Both single-pulse ($B = 0$) and double-pulse ($B \neq 0$) fields are considered.

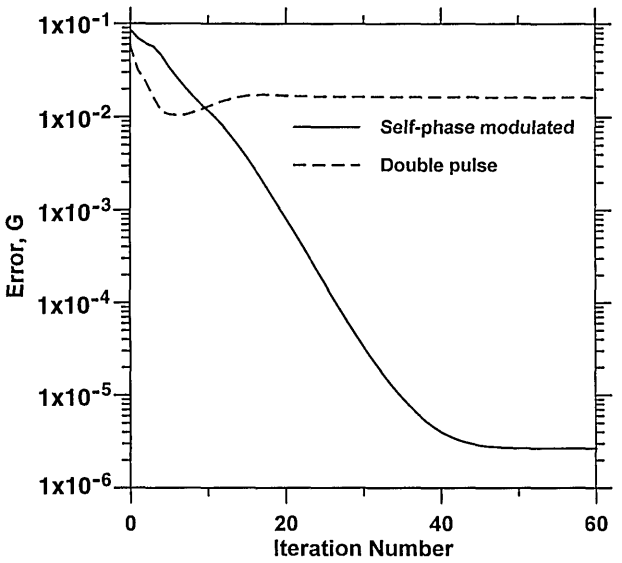

Fig. 2. Error as a function of iteration number for two different pulses. Both pulses were computed on a 64-element array (FROG trace of 64 × 64 pixels). The basic FROG algorithm easily converges to the self-phase-modulated pulse, which has a FWHM of 10 and $Q = 4$ [see Eq. (7)]. The algorithm stagnates, however, for the double pulse, which has a FWHM of 6, with $B = 1$ and $D = 12$, as defined in Eq. (8), and no phase distortion ($A = Q = 0$).

Instead, the error remains at a high value, and the algorithm stagnates, i.e., the current guess for $E(t)$ ceases to change, despite the high error value.

The cause of the failure of the algorithm in the double-pulse case is not easily discernible. We can use an initial guess for $E(t)$ of the double pulse that is exactly correct, except that the intensity of one of the peaks is reduced by 1%. Even with this nearly correct starting point the algorithm will diverge and move to a stagnating solution with a higher error G than it began with. The cause of this problem seems to be associated with the common problem of striping[9] in the usual phase-retrieval problem. In the cases in which the FROG algorithm stagnates while attempting to retrieve a double pulse, we note the presence of anomalous stripes in the reconstructed FROG trace. In normal phase retrieval, striping in the image domain is associated with stagnation of the algorithm at high error values. The striping problem has been a particularly vexing one (for both FROG and standard phase retrieval), and the solutions used by the phase-retrieval community are either inapplicable or ineffective when they are applied to the FROG algorithm.

We can consider the basic FROG (iterative Fourier transform) algorithm as a map M that maps one electric field $E(t)$ as a new field, $E_{n+1} = M(E_n)$.[10] In this case we see that the correct solution for $E(t)$ is a fixed point of the map described by the basic FROG algorithm: the field is not changed after one iteration of the algorithm.

For most pulses and for the double pulse in which the intensity of the second peak is less than half that of the primary peak the fixed point representing the correct solution for the electric field is a stable fixed point. Even more importantly, it appears to be an attractor for the motion of the system: nearby points in phase space converge on the fixed point under repeated iterations of the map. For a double pulse of equal intensities the correct solution is an unstable fixed point and is therefore no longer an attractor. Thus even small perturbations from the fixed point cause the solution to diverge, as we discussed above. In this case a separate stable, attractive fixed point appears: the stagnating solution to which the basic FROG algorithm evolves. For double pulses with intensity ratios of close to 2:1 (within a few percent) the electric field cycles through a repeating set of values: the attractor of the map is a periodic orbit.

The exact reason for the transition to instability of the fixed point representing the correct solution or the appearance of an attractor representing an incorrect solution is not understood and is an interesting topic for further research. It is clear from these considerations that the double pulse represents a particularly pathological case for the basic FROG algorithm. Fortunately, we have developed techniques that permit us to overcome this problem. We now discuss those techniques.

5. INTENSITY CONSTRAINT

The failure of the basic FROG algorithm to converge to the double pulse is apparently due to underconstraint—the constraints that have been used [the experimental FROG trace and Eqs. (1) and (4)] are insufficient to force convergence to the correct field. Therefore the use of some additional constraint may improve the performance of the algorithm. Fortunately, such a constraint is available in FROG.

We have achieved considerable success with what we call an intensity constraint. The intensity constraint is formed by the use of additional information from the signal field, in this case the form of the gate (which is the intensity envelope of the pulse). The use of this additional constraint permits us to get much closer to the correct solution in cases in which the basic FROG algorithm does not converge.

In order to generate the gate function from the signal field, we note that a generic form of Eqs. (1) and (2) is

$$E_{\mathrm{sig}}(\omega, \tau) = \int_{-\infty}^{\infty} \mathrm{d}t\, F(t) G(t - \tau) \exp(i\omega t), \tag{9}$$

where $F(t)$ is the field and $G(t)$ is the gate. To extract the gate function, we integrate with respect to ω to yield

$$\int_{-\infty}^{\infty} \mathrm{d}\omega\, E_{\mathrm{sig}}(\omega, \tau) = F(0) G(-\tau). \tag{10}$$

The integration of the signal field thus yields a time-reversed version of the gate function (without an inverse Fourier transform), which in the case of polarization-gate FROG is just the intensity envelope of the pulse [integrating Eq. (9) with respect to τ yields $E(\omega)$, which we can inverse Fourier transform to yield $E(t)$, as in the discussion following Eq. (4)]. To apply the intensity constraint, we replace the intensity of the current guess for $E(t)$ with the (time-reversed) intensity derived by integration of $E'_{\mathrm{sig}}(\omega, \tau)$ with respect to ω, while leaving the phase unchanged.

Repeated application of the intensity constraint alone leads to large errors in the phase of the current-guess field, as well as noise in the wings of the field. We therefore alternate between the basic FROG algorithm and the

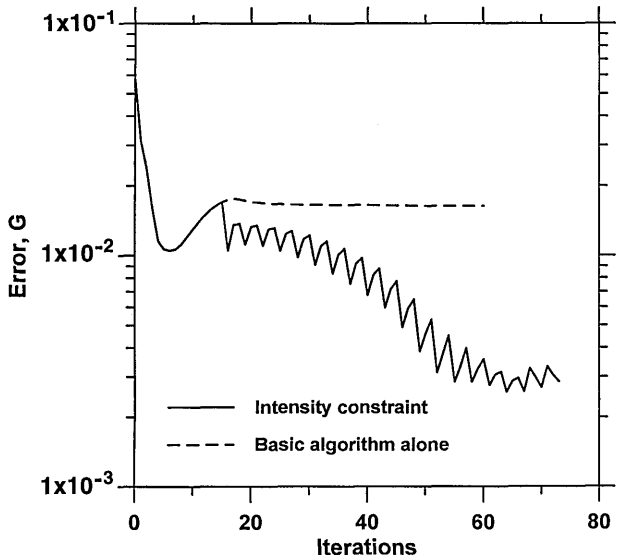

Fig. 3. Use of the intensity-constraint method enables the algorithm to make greater progress on the double pulse. The pulse was computed on a 64-element array, with a FWHM of 6 and $B = 1$, $D = 12$, and $A = Q = 0$, as defined in Eq. (8). The fluctuations are due to the method of implementation: two iterations with the intensity constraint followed by one iteration of the basic FROG algorithm.

use of the intensity constraint when applying this method. We find empirically that two iterations with the intensity constraint followed by one iteration with the basic FROG algorithm works quite well. The error G actually gets larger during the application of the intensity constraint and is reduced with the application of the basic FROG algorithm, but the overall result is a much smaller error than that attainable with the basic FROG algorithm alone.

This intensity-constraint method was applied with the use of a FROG trace generated by a double pulse of equal intensities as an input. These results are seen in Fig. 3, which shows the error as a function of iteration number for both the basic FROG algorithm and the intensity-constraint method. The basic FROG algorithm was applied until a minimum error level was reached, and then the alternation of the basic FROG algorithm and the intensity constraint described above was used. We see that this combination reaches much smaller error values than the basic FROG algorithm alone. However, for the double pulse the algorithm still does not converge to a stable, low-error solution. Additional techniques are required and are discussed below.

6. OVERCORRECTION

Another modification of the basic FROG algorithm is what we will call overcorrection. In the basic FROG algorithm we replace the magnitude of the signal field $E_{\text{sig}}(\omega, \tau)$ generated by the current guess for $E(t)$ with the magnitude of the experimentally generated FROG data, as described by Eq. (3). This replacement corrects the magnitude of $E_{\text{sig}}(\omega, \tau)$, that is, it increases the magnitude where it is too small and decreases it where it is too large, yielding $E'_{\text{sig}}(\omega, \tau)$ with a magnitude that is equal to that of the actual FROG trace.

One might imagine that overcorrecting for the errors between $|E_{\text{sig}}(\omega, \tau)|^2$ and $I_{\text{FROG}}(\omega, \tau)$ could speed the convergence of the algorithm. For example, we could add an overcorrection term that is proportional to the deviation between the two FROG traces, such as

$$E'_{\text{sig}}(\omega, \tau) = \frac{E_{\text{sig}}(\omega, \tau)}{|E_{\text{sig}}(\omega, \tau)|}[I_{\text{FROG}}(\omega, \tau)]^{1/2} \times \left[1 + \frac{I_{\text{FROG}}(\omega, \tau) - |E_{\text{sig}}(\omega, \tau)|^2}{|E_{\text{sig}}(\omega, \tau)|^2}\right]. \quad (11)$$

The inclusion of the factor in square brackets has the effect of overcorrecting E'_{sig}: where the squared magnitude of E_{sig} is smaller than that of I_{FROG}, the new squared magnitude of E'_{sig} will be larger than that of I_{FROG} and vice versa.

Equation (11) becomes

$$E'_{\text{sig}}(\omega, \tau) = \frac{E_{\text{sig}}(\omega, \tau)}{|E_{\text{sig}}(\omega, \tau)|}[I_{\text{FROG}}(\omega, \tau)]^{1/2}\left[\frac{I_{\text{FROG}}(\omega, \tau)}{|E_{\text{sig}}(\omega, \tau)|^2}\right] = E_{\text{sig}}(\omega, \tau)\left\{\frac{[I_{\text{FROG}}(\omega, \tau)]^{1/2}}{|E_{\text{sig}}(\omega, \tau)|}\right\}^3. \quad (12)$$

We can generalize this replacement step to

$$E'_{\text{sig}}(\omega, \tau) = E_{\text{sig}}(\omega, \tau)\left|\frac{[I_{\text{FROG}}(\omega, \tau)]^{1/2}}{E_{\text{sig}}(\omega, \tau)}\right|^{\alpha}, \quad (13)$$

where α is an adjustable exponent. In the basic FROG algorithm, $\alpha = 1$. This approach has also been used in standard phase retrieval and blind deconvolution algorithms.[11]

We find that values of α slightly larger than unity can speed convergence. Values larger than roughly 1.5 can cause the algorithm to become unstable, however. The value of α at which the instability begins is dependent on the size of the error G: the smaller that G is (the closer to convergence), the larger is the value of α that can safely be used. The reason for the instability can be easily seen. A field with a large error G by definition has large deviations from the correct FROG trace. When the value of α is high, a large error will lead to a large correction. If the correction is too large, this will lead to even greater errors in the field, which then lead to larger corrections, etc. An unstable positive feedback cycle is created, and the error (and field) diverges.

Figure 4 shows the effect of different values of α on the convergence of the algorithm for a self-phase-modulated pulse with $Q = 4$. We see that, although increasing α to approximately 1.5 speeds convergence, values of 2.0 or larger cause the algorithm to diverge. Larger values of α can be used, however, if the basic FROG algorithm is permitted to reduce the error before the larger α are used. Alternatively, one can use a value of α that begins at unity and slowly increases with iteration number. Using such a scheme, we have seen the algorithm behave in a stable manner with values of α larger than 3.

Increasing α beyond unity can serve to increase the speed and the level of convergence for troublesome pulses with complicated phase profiles that converge slowly under the basic FROG algorithm. While it is generally quite helpful, this scheme unfortunately does not seem to

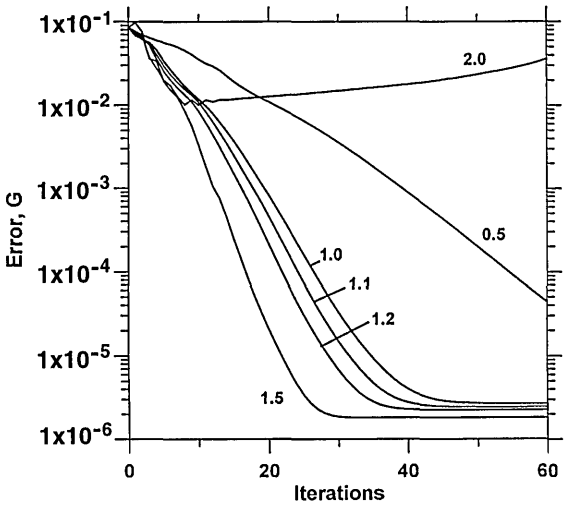

Fig. 4. Effect of different values of the exponent α in the overcorrection method. Although raising the value of α increases the rate of convergence, too high a value can cause the algorithm to become unstable. The pulse was self-phase modulated, with a FWHM of 10 and $Q = 4$ on a 64-element grid.

be able to overcome the stagnation problems associated with the double pulse.

7. MINIMIZATION TECHNIQUES

Although the above techniques yield significantly improved results, this improved algorithm still stagnates for some pulses, including the double pulse. The inclusion of another technique, multidimensional minimization, permits us to achieve convergence in essentially all cases of interest, including the double pulse.

Minimization involves considering the FROG error G to be a single-valued function of $2N$ variables. The $2N$ variables are the values of the real and imaginary parts of the electric field at each of the N sampling points of the array that holds the field. Thus we can consider the pulse-retrieval problem as a multidimensional minimization, where the goal is to minimize the FROG error G.[6]

In a multidimensional minimization problem considerable computational effort can often be saved if we can calculate the gradient of the function at an arbitrary point. Luckily, in FROG we are able to do this. We first define a new error function,

$$H = \sum_{\omega,\tau=1}^{N} [I_{\mathrm{FROG}}(\omega, \tau) - |E_{\mathrm{sig}}(\omega, \tau)|^2]^2, \tag{14}$$

which is related to our usual error function G through the simple relation $H = G^2N^2$. The signal field is defined as

$$E_{\mathrm{sig}}(\omega, \tau) = \sum_{t=1}^{N} E(t)|E(t - \tau)|^2 \exp(i\omega t), \tag{15}$$

where $\omega = 2\pi f/N$ and f takes on integer values when sums over ω are performed.

Our task is to take the derivative of H for every point in the field $E(t)$. We need to take the derivative separately for the real and imaginary parts of $E(t)$. After some algebra we find that

$$\begin{aligned}\frac{dH}{dE^r(t_0)} = & -4\sum_{\omega,\tau}^{N}[I_{\mathrm{FROG}}(\omega, \tau) - |E_{\mathrm{sig}}(\omega, \tau)|^2] \\ & \times \mathrm{Re}(\{|E(t_0 - \tau)|^2 \exp(i\omega t_0) \\ & + 2E^r(t_0)E(t_0 + \tau)\exp[i\omega(t_0 + \tau)]\} \\ & \times E^*_{\mathrm{sig}}(\omega, \tau)),\end{aligned} \tag{16}$$

and for the imaginary part of the field we have

$$\begin{aligned}\frac{dH}{dE^i(t_0)} = & -4\sum_{\omega,\tau}^{N}[I_{\mathrm{FROG}}(\omega, \tau) - |E_{\mathrm{sig}}(\omega, \tau)|^2] \\ & \times \mathrm{Re}(\{i|E(t_0 - \tau)|^2 \exp(i\omega t_0) \\ & + 2E^i(t_0)E(t_0 + \tau)\exp[i\omega(t_0 + \tau)]\} \\ & \times E^*_{\mathrm{sig}}(\omega, \tau)),\end{aligned} \tag{17}$$

where we have used the relations

$$\frac{dE(t)}{dE^r(t_0)} = \delta_{t,t_0} \qquad \frac{dE(t)}{dE^i(t_0)} = i\delta_{t,t_0}, \tag{18}$$

where $\delta_{i,j}$ is unity for $i = j$ and zero for $i \neq j$.

Now that we are in possession of the gradient of the error function, we can apply a minimization technique. We have used the standard Fletcher–Reeves and Polak–Ribiere methods,[12] which are conjugate gradient methods. These involve a series of one-dimensional minimizations along directions that are selected with the aid of the gradient. The reader is referred to Ref. 12 for further details.

In contrast to the iterative Fourier transform methods discussed above, the minimization routine will necessarily always act to reduce the error. However, it cannot distinguish between global and local minima. We find that the multidimensional surface of $H(E)$ is fraught with a plethora of local minima, and therefore the minimization routine needs a fairly good initial guess in order to ensure convergence. Even so, we cannot determine whether a given input will converge to the global minimum or get caught in one of the numerous local minima. Therefore we must expect that in the general case the minimization routine will become trapped in local minima.

We can imagine that a different error function with the same global minimum might have different local minima. Using such a function might assist in convergence to the global minimum. Therefore, in order to kick the minimization routine out of local minima, we alter the surface of the error function by constructing a modified error,

$$H_W = \sum_{\omega,\tau}^{N}\left[\frac{I_{\mathrm{FROG}}(\omega, \tau) - |E_{\mathrm{sig}}(\omega, \tau)|^2}{I_{\mathrm{FROG}}(\omega, \tau)}\right]^2. \tag{19}$$

(This error function has the effect of emphasizing the areas of the FROG trace that are small in intensity, and one of the resulting effects is that it helps to determine the wings of the pulse.) The local minima associated with this new error function hopefully are located at different positions, so that switching to this new error function can serve to kick the minimization routine out of a local minimum and send it closer to the global minimum. In practice, we sometimes find that this is effective, whereas in other cases switching to the new error function does

not serve to free the minimization routine from the local minimum.

Because of the abundance of local minima, a good initial guess is required for the minimization scheme. We find that, by using the combination of the basic FROG algorithm, the intensity constraint, and overcorrection, we can obtain an initial guess for the minimization routine that permits it to converge to a satisfactory solution for the vast majority of cases.

8. CONTROL OF THE COMPOSITE ALGORITHM

We are now in the possession of several techniques—the basic FROG algorithm, the intensity-constraint method, the overcorrection technique, and the multidimensional minimization techniques—that permit us to retrieve an electric field from the FROG trace that it created. The question now is, how can we combine these techniques to maximize the benefits of each?

We have created a composite algorithm that combines all these techniques in a way that should converge for essentially all the possible pulse types of interest without operator intervention. This composite algorithm is able to monitor its own progress, decide when a particular technique is failing to converge, and select a new technique to try.

In order to monitor its progress, our composite algorithm examines the error G after every iteration of each technique used. It stores the error for the previous nine iterations, as well as the smallest error, $G_{\min}$, achieved and its associated field, $E_{\min}(t)$.

The algorithm continues using a selected technique until one of two conditions is detected: convergence or stagnation. Convergence is defined as an error G of less than 0.0001. (Note that we are dealing with numerically generated input pulses here, i.e., noise-free pulses. The effect of the noise inherent in real experimental data has not yet been fully illuminated.) We define stagnation by the condition that the error G has decreased by less than 0.5% from the value nine iterations prior to that. When the algorithm switches to a new technique, it resets the current guess for the field to $E_{\min}(t)$, which is that field which generated the lowest previous error. In this way techniques that occasionally diverge (increase the error) will not reduce the efficiency of the composite algorithm.

The flow of the composite algorithm is diagrammed in Fig. 5. After using all the iterative Fourier transform methods, the algorithm then applies the minimization techniques. If convergence is not yet achieved, the algorithm enters a loop in which it repeatedly applies the iterative Fourier transform with intensity-constraint method and minimization techniques. We now discuss the details of each of these stages.

The basic FROG algorithm, which uses the iterative Fourier transform, is the simplest, most robust, and quickest of the methods available. It generally suffices in most instances, excluding those in which stagnation is a problem, as in the double-pulse case discussed in Section 4. We therefore begin our composite algorithm with the basic FROG algorithm. If stagnation is detected, the algorithm moves on to the overcorrection method. The value of α used increases with the number of iterations k of the overcorrection method through

$$\alpha = (1.1)^{1+k/5}. \tag{20}$$

However, there may be room for further optimization of this function, as it is heuristically determined. If the value of α increases too quickly, the technique may become unstable and diverge, as we discussed in Section 6. If α increases too slowly, however, its effect will be minimal at first, and the self-regulating portion of the composite algorithm may decide that convergence is occurring too slowly and skip to the next method.

If the algorithm has not yet converged, we next apply the intensity-constraint method. During the application of the intensity-constraint method the error fluctuates quite a bit (see Fig. 3), so we must exercise caution when testing for convergence or stagnation. Remember that, in the intensity-constraint method, the algorithm performs two iterations using the intensity constraint [Eq. (10)], followed by one iteration of the basic FROG algorithm, so that the period of these fluctuations is three iterations. We therefore average the error over three iterations before comparing it with the error nine iterations previously (which has also been averaged over three iterations). Also, if the field with the lowest error $G_{\min}$ occurred within the last nine iterations, we permit the intensity-constraint portion of the algorithm to continue, even if it meets the other criteria for stagnation. These considerations allow for the often complex dynamics of the intensity-constraint method.

If the intensity-constraint method stagnates, we then use the minimization schemes. These are computationally more expensive than the iterative Fourier transform methods (order N^3 versus $N^2 \ln N$ for the iterative Fourier transform techniques). We begin with the Polak–Ribiere method. When using the minimization schemes, the algorithm keeps track of the error values of only the previous four iterations.

If the Polak–Ribiere method stagnates, we switch to the new error function H_W of Eq. (19). This error function presents us with some unique problems, since it often happens that H increases as H_W decreases, at least initially. In the case in which the algorithm is converging

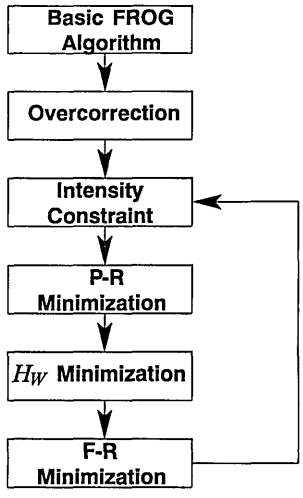

Fig. 5. Flow chart showing the control of the composite FROG algorithm. P-R stands for Polak–Ribiere, and F-R stands for Fletcher–Reeves.

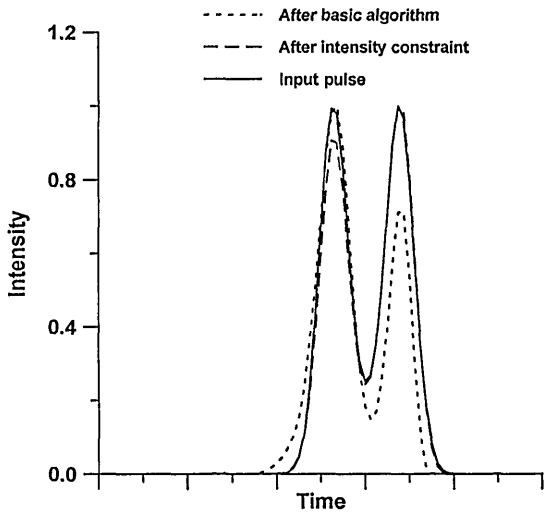

Fig. 6. Intensity profiles of the field with the lowest error produced by the basic FROG algorithm and the intensity-constraint method in the case of a double pulse. The intensity-constraint method gets significantly closer to the correct answer. The pulse had a FWHM of 6, with $B = 1$ and $D = 12$ [see Eq. (8)], and no phase distortion.

to the global minimum, this is rarely a problem, as convergence is usually swift and smooth. In the rare case in which the minimization technique operating on the error H is stagnated in a particularly deep local minimum, although the algorithm may make progress in minimizing H_W, the new field may be lost because the error H increased (causing the self-monitoring portion of the algorithm to discard that field). To guard against this occurrence, we permit the new field generated by minimizing H_W to be passed to the next routine, even if H increased, on alternate passes through this portion of the minimization technique.

If this method stagnates, the algorithm switches back to the original error function H but uses the Fletcher–Reeves method. The Fletcher–Reeves method differs from the Polak–Ribiere method in the way that it generates new directions in which to minimize the error function.[12] Finally, if this fails, the algorithm switches back to the intensity-constraint method, and the cycle begins again, as Fig. 5 shows.

This composite algorithm as described above converges for a much broader class of pulses than the basic FROG algorithm. Figure 6 shows the performance of the basic FROG algorithm, the intensity-constraint method, and the entire composite algorithm in the case of a double pulse with two equal-intensity peaks. The intensity-constraint method gets significantly closer to convergence than the basic FROG algorithm ($G = 0.00226$ versus $G = 0.0105$), but the minimization routines are required for complete convergence. The field retrieved by the full composite algorithm in Fig. 6 is visually indistinguishable from the field used to generate the FROG trace.

Nearly all pulses with complicated intensity and/or phase profiles that we tried were retrieved by the composite algorithm, and some examples are shown in Fig. 7. These waveforms are sums of four individual Gaussian pulses, each with varying height and width and various phase distortions. All these pulses were successfully retrieved with the composite algorithm.

The composite algorithm was also tested on pulses with abrupt jumps in phase. The algorithm was able to re-

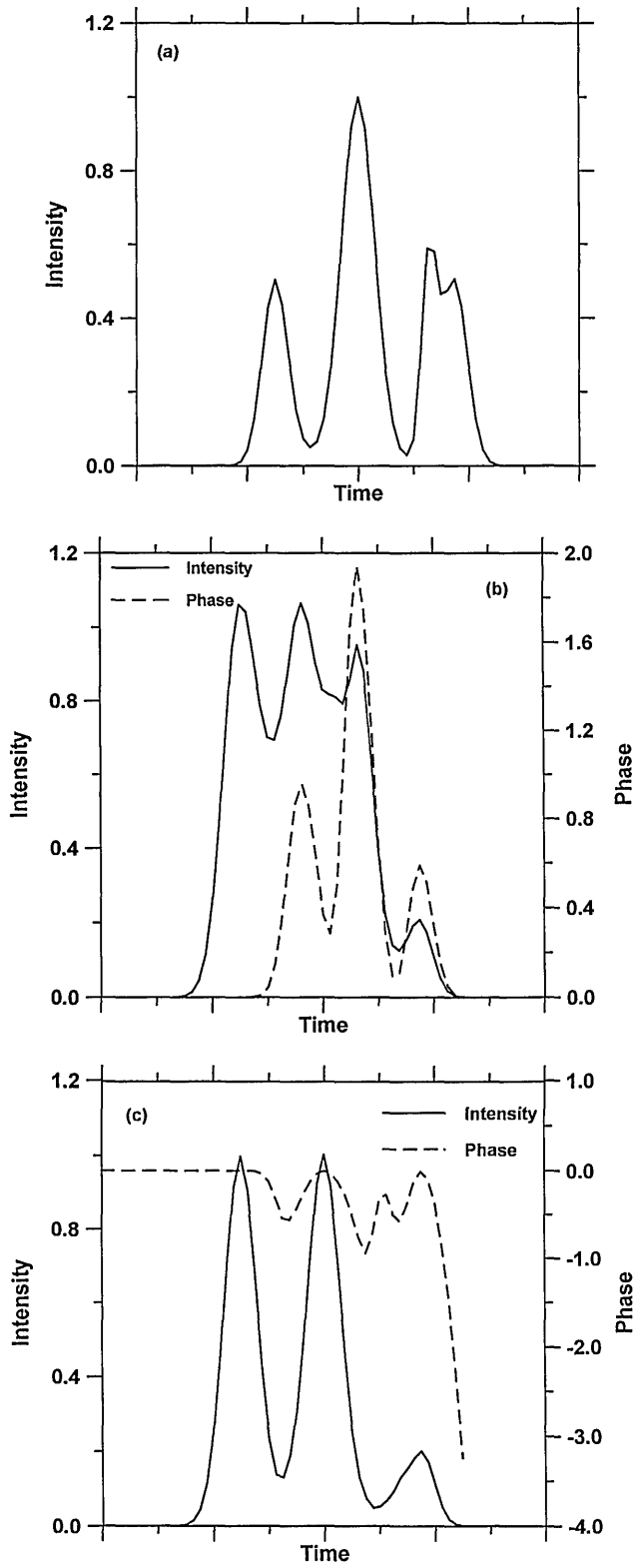

Fig. 7. Intensity and phase of three highly structured pulses that the composite algorithm succeeded in retrieving. All three pulses are the sum of four individual Gaussian pulses, with varying widths, heights, and amounts of phase distortion. In (a) there is no phase distortion, in (b) each pulse has a varying amount of self-phase modulation, and in (c) each pulse has a varying amount of linear chirp. These complicated pulses could not be retrieved from the basic FROG algorithm alone.

trieve these pulses as long as the phase jump was not too sharp (less than two array elements wide) and as long as the phase change between adjacent points was less than roughly $0.5\pi-0.3\pi$. Also, the composite algorithm had no trouble with phase ramps, where the trailing half of the pulse has a linear phase ramp (frequency shift), again assuming that the phase shift between adjacent points was not too large. In the cases in which the composite algorithm did not converge initially, we found that, after the point density is increased, the composite algorithm converges without difficulty.

9. CLEVER TRICKS THAT FAILED

The case of the double pulse is an extremely recalcitrant one for the basic FROG algorithm, and before the composite algorithm mentioned in Section 8 solved the problem, most of our initial attempts to solve this problem failed. Many of these initial techniques attempted were taken from the phase-retrieval literature and therefore had a high *a priori* probability of success. We wish to document these techniques in order that others need not duplicate our failures. Also, these tricks may be effective in other problems that bear a resemblance to phase retrieval or FROG.

As reported in Ref. 1, application of support constraints[7] was found to be ineffective in FROG. We tried arbitrary triangular or rectangular support areas in the image domain [created by the inverse Fourier transformation of $E_{\text{sig}}(t, \tau)$ with respect to τ], as well as supports in the $E(t)$ domain, where we used static supports as well as ones that narrowed or widened with iteration number. Pulses that converged in the basic FROG algorithm converged more slowly in the presence of the support constraint, while pulses like the double pulse remained unyielding.

We also tried mimicking the hybrid input–output algorithm.[6] Instead of the application of data indicated by Eq. (3), we used the replacement

$$E'_{\text{sig}}(\omega, \tau) = \frac{E_{\text{sig}}(\omega, \tau)}{|E_{\text{sig}}(\omega, \tau)|}\{(1-\beta)[I_{\text{FROG}}(\omega, \tau)]^{1/2} + \beta|E_{\text{sig}}(\omega, \tau)|\}, \quad (21)$$

where $\beta < 1$. This also was ineffective, as was applying the same technique in the field domain.

Another disappointing technique was the application of a spectral constraint. Since the spectrum is an experimentally accessible quantity, it seemed reasonable to attempt to use it to aid convergence. After a new field was generated by the integration of Eq. (4), it was Fourier transformed and the magnitude of its spectrum was replaced with that of the original pulse. Surprisingly, this had the effect of slowing down the convergence for well-behaved pulses and did not help in the case of the double pulse.

10. CONCLUSION

An improved algorithm for retrieving ultrashort pulses from their FROG traces was discussed. This new algorithm converges in essentially all reasonable cases, including those in which the original basic FROG algorithm did not. We described the modifications to the original algorithm, including the use of an intensity constraint, an overcorrection technique, and a multidimensional minimization method, and showed examples of complex field profiles that were retrieved with the new algorithm. We also presented several examples of promising techniques that did not aid the convergence of the algorithm.

While the basic FROG algorithm described in a previous publication converged for most commonly occurring ultrashort pulses, use of the composite algorithm presented here enables convergence for essentially all relevant pulses that we have been able to construct. As a result, the composite algorithm should provide a practical laboratory tool for the quantitative determination of an ultrashort pulse's intensity and phase.

ACKNOWLEDGMENTS

We would like to thank Lisa Bernstein for illuminating discussions on dynamical systems. The authors also acknowledge the support of the U.S. Department of Energy, Office of Basic Energy Sciences, Chemical Sciences Division.

REFERENCES

1. R. Trebino and D. J. Kane, "Using phase retrieval to measure the intensity and phase of ultrashort pulses: frequency-resolved optical gating," J. Opt. Soc. Am. A **10,** 1101–1111 (1993).
2. D. J. Kane and R. Trebino, "Single-shot measurement of the intensity and phase of an arbitrary ultrashort pulse by using frequency-resolved optical gating," Opt. Lett. **18,** 823–825 (1993).
3. R. A. Altes, "Detection, estimation, and classification with spectrograms," J. Acoust. Soc. Am. **67,** 1232–1246 (1980).
4. K. W. DeLong, R. Trebino, and D. J. Kane, "Comparison of ultrashort-pulse frequency-resolved-optical-gating traces for three common beam geometries," J. Opt. Soc. Am. B (to be published).
5. D. J. Kane, A. J. Taylor, R. Trebino, and K. W. DeLong, "Single-shot measurement of the intensity and phase of a femtosecond UV laser pulse with frequency-resolved optical gating," Opt. Lett. **18,** 1061–1063 (1994).
6. J. R. Fienup, "Phase retrieval algorithms: a comparison," Appl. Opt. **21,** 2758–2769 (1982).
7. J. R. Fienup, "Reconstruction of a complex-valued object from the modulus of its Fourier transform using a support constraint," J. Opt. Soc. Am. A **4,** 118–123 (1987).
8. J. R. Fienup and A. M. Kowalczyk, "Phase retrieval for a complex-valued object by using a low-resolution image," J. Opt. Soc. Am. A **7,** 450–458 (1990).
9. J. R. Fienup and C. C. Wackerman, "Phase-retrieval stagnation problems and solutions," J. Opt. Soc. Am. A **3,** 1897–1907 (1986).
10. A. J. Lichtenberg and M. A. Lieberman, *Regular and Stochastic Motion,* Vol. 38 of Applied Mathematical Sciences (Springer-Verlag, New York, 1983).
11. J. H. Seldin and J. R. Fienup, "Iterative blind deconvolution algorithm applied to phase retrieval," J. Opt. Soc. Am. A **7,** 428–433 (1990).
12. W. H. Press, W. T. Vetterling, and S. A. Teukolsky, *Numerical Recipes in C,* 2nd ed. (Cambridge U. Press, Cambridge, UK, 1992), pp. 420–425.