



High-speed “multi-grid” pulse-retrieval algorithm for frequency-resolved optical gating

RANA JAFARI,* AND RICK TREBINO

School of Physics, Georgia Institute of Technology, 837 State St NW, Atlanta, GA 30332, USA

*rjafari7@gatech.edu

Abstract: We use an algorithmic technique called “multi-grid” to improve the speed of convergence of the cross-correlation frequency-resolved-optical-gating (XFROG) pulse-retrieval algorithm for very complex pulses. The multi-grid approach uses a smaller trace ($N/4 \times N/4$) drawn from the original $N \times N$ trace for initial iterations, yielding poorer resolution and range, but proceeding ~ 16 times faster for such iterations. The pulse field rapidly retrieved from this smaller array then provides the initial guess for the larger, full array, significantly reducing the number of iterations required on the full array. We first find that, for simple pulses and their resulting simple traces, the original generalized-projections FROG and XFROG algorithms already converge in less time than is required to plot the retrieved pulse, so speed improvements for them appear irrelevant in general. Considering therefore only complex pulses and their resulting complex traces, we adapted the multi-grid algorithm to XFROG, the technique used for complex pulses whenever possible. We show that extending multi-grid to even smaller arrays is not helpful, but intermediate-size arrays of $N/2 \times N/2$ are, further reducing the number of iterations on the full array and further decreasing convergence time. We obtain a factor of ~ 7 improvement in speed for very complex pulses with time-bandwidth products of 50 to 90. This approach does not require modifications to the algorithm itself and so can be used in conjunction with essentially all FROG algorithms for improved speed. And it retains FROG’s ability to determine the pulse-shape stability in multi-shot measurements.

© 2018 Optical Society of America under the terms of the [OSA Open Access Publishing Agreement](#)

OCIS codes: (320.7100) Ultrafast measurements; (100.5070) Phase retrieval.

References and links

1. R. Trebino, *Frequency-Resolved Optical Gating: The Measurement of Ultrashort Laser Pulses* (Kluwer Academic Publishers, Boston, 2002).
2. K. W. DeLong, C. L. Ladera, R. Trebino, B. Kohler, and K. R. Wilson, “Ultrashort-pulse measurement using noninstantaneous nonlinearities: Raman effects in frequency-resolved optical gating,” *Opt. Lett.* **20**(5), 486–488 (1995).
3. H. G. S. Linden and J. Kuhl, “XFROG: new method for amplitude and phase characterization of ultraweak ultrashort pulses,” *Phys. Status Solidi, B Basic Res.* **206**(1), 119–124 (1998).
4. K. W. DeLong, D. N. Fittinghoff, R. Trebino, B. Kohler, and K. Wilson, “Pulse retrieval in frequency-resolved optical gating based on the method of generalized projections,” *Opt. Lett.* **19**(24), 2152–2154 (1994).
5. T. Bendory, P. Sidorenko, and Y. C. Eldar, “On the Uniqueness of FROG Methods,” *IEEE Signal Process. Lett.* **24**(5), 722–726 (2017).
6. D. N. Fittinghoff, K. W. DeLong, R. Trebino, and C. L. Ladera, “Noise sensitivity in frequency-resolved optical-gating measurements of ultrashort pulses,” *J. Opt. Soc. Am. B* **12**(10), 1955–1967 (1995).
7. L. Xu, E. Zeek, and R. Trebino, “Simulations of frequency-resolved optical gating for measuring very complex pulses,” *J. Opt. Soc. Am. B* **25**(6), A70–A80 (2008).
8. X. Gu, L. Xu, M. Kimmel, E. Zeek, P. O’Shea, A. P. Shreenath, R. Trebino, and R. S. Windeler, “Frequency-resolved optical gating and single-shot spectral measurements reveal fine structure in microstructure-fiber continuum,” *Opt. Lett.* **27**(13), 1174–1176 (2002).
9. M. Rhodes, G. Steinmeyer, J. Ratner, and R. Trebino, “Pulse-shape instabilities and their measurement,” *Laser Photonics Rev.* **7**(4), 557–565 (2013).
10. M. Rhodes, Z. Guang, and R. Trebino, “Unstable and Multiple Pulsing Can Be Invisible to Ultrashort Pulse Measurement Techniques,” *Appl. Sci.* **7**(1), 40 (2017).

11. J. W. Nicholson, F. G. Omenetto, D. J. Funk, and A. J. Taylor, "Evolving FROGS: phase retrieval from frequency-resolved optical gating measurements by use of genetic algorithms," *Opt. Lett.* **24**(7), 490–492 (1999).
12. P. Sidorenko, O. Lahav, Z. Avnat, and O. Cohen, "Ptychographic reconstruction algorithm for frequency-resolved optical gating: super-resolution and supreme robustness," *Optica* **3**(12), 1320–1330 (2016).
13. D. Spangenberg, E. Rohwer, M. H. Brüggemann, and T. Feurer, "Ptychographic ultrafast pulse reconstruction," *Opt. Lett.* **40**(6), 1002–1005 (2015).
14. M. J. Stimson, D. J. Ulness, J. C. Kirkwood, G. S. Boutis, and A. C. Albrecht, "Noisy-light correlation functions by frequency resolved optical gating," *J. Opt. Soc. Am. B* **15**(2), 505–514 (1998).
15. P.-Y. Wu, H.-H. Lu, C.-Z. Weng, Y.-H. Chen, and S.-D. Yang, "Dispersion-corrected frequency-resolved optical gating," *Opt. Lett.* **41**(19), 4538–4541 (2016).
16. G. Stibenz and G. Steinmeyer, "Interferometric frequency-resolved optical gating," *Opt. Express* **13**(7), 2617–2626 (2005).
17. G. P. Agrawal, *Nonlinear Fiber Optics* (Academic Press, Boston, 1995).
18. G. Genty, S. Coen, and J. M. Dudley, "Fiber supercontinuum sources (Invited)," *J. Opt. Soc. Am. B* **24**(8), 1771–1785 (2007).
19. C. W. Siders, J. L. W. Siders, F. G. Omenetto, and A. J. Taylor, "Multipulse interferometric frequency-resolved optical gating," *IEEE J. Quantum Electron.* **35**(4), 432–440 (1999).

1. Introduction

The well-known ultrashort pulse measurement technique, frequency-resolved optical gating (FROG), operates in the time-frequency domain [1], utilizing a two-dimensional array of data, $I_{FROG}(\omega, \tau)$, acquired from any spectrally resolved nonlinear-optical gating arrangement. The measured "FROG trace" can be described by the following relation:

$$I_{FROG}(\omega, \tau) \propto \left| \int_{-\infty}^{\infty} E_{sig}(t, \tau) \exp(-i\omega t) dt \right|^2, \quad (1)$$

where E_{sig} is the signal field generated in any instantaneous or even non-instantaneous nonlinear optical process [2] in which one pulse time- or frequency-gates another. In self-referenced versions of FROG, $E_{sig}(t, \tau)$ is a function of $E(t)$ and a variably delayed replica of it, $E(t-\tau)$. If a known gate pulse is available, $E_{sig}(t, \tau)$ becomes a function of the reference pulse and the unknown pulse in a variant called cross-correlation FROG (XFROG) [3].

FROG's well-known generalized-projections (GP) algorithm [4] reliably and quickly retrieves the temporal intensity and phase of arbitrary pulses. It involves Fourier-transforming between the time (t) and frequency (ω) domains and applying the relevant constraint in the respective domain (the functional form of E_{sig} or the measured trace), to retrieve the intensity and phase of the pulse from the FROG trace and to do so essentially uniquely [1, 5].

While the GP algorithm has proven quite effective in a wide range of cases [6, 7], retrieval can be quite slow for very complex pulses. For example, retrieving an extremely complex continuum pulse with a time-bandwidth product (TBP) of ~ 4000 from its 4096×4096 XFROG trace required a few hours [8]. Of course, computers were considerably slower then (2001). Also, it is not clear what role the pulse train's extremely unstable pulse shapes and the resulting discrepancies between measured and retrieved traces (a unique and useful feature of FROG to reveal pulse-shape instability [9, 10]) played in this particular case. Nevertheless, the algorithm could use a speed increase for complex pulses with such large traces.

Since FROG's introduction, several additional algorithmic methods have been introduced for it, for example, genetic [11] and ptychographic [12, 13] methods for various reasons, including improved speed. Also, additional algorithms and variations on GP have been developed for various versions of FROG [14–16]. However, to our knowledge, no algorithm yet combines the speed and reliability of the original GP algorithm, especially for complex pulses.

Inspired by recent discussions of algorithm speed, we considered trying to speed up the GP algorithm for its most common application: simple pulses measured by second-harmonic-generation (SHG) FROG. However, we quickly found that the original GP algorithm, coded in C++ on an inexpensive laptop, retrieves pulses from 32×32 traces (the optimal trace size

for such measurements) in ~ 0.05 s, which is less time than that required to plot the resulting pulse. XFROG is even faster (0.02s). We conclude that there is not much to be gained by attempts to speed up the FROG algorithm for simple pulses, the goal having already been achieved by massive improvements in computer CPU speed over the past two-and-a-half decades (i.e., by doing nothing). If additional speed is desired, buying a faster computer and/or waiting a year or two for the next generation of processors seem the best strategies.

On the other hand, with recent increases in applications of complex pulses, such as shaped pulses [17] and continuum [18], a faster retrieval algorithm for XFROG, the specific FROG technique used for such pulses—a complex pulse is best measured using a simple one, rather than a complex one—is actually needed. As a result, in this work, we attempt to reduce the retrieval time for large traces of complex pulses by applying to the standard GP XFROG algorithm a “multi-grid” approach, first introduced by Siders, et al., for a related FROG technique with complex traces, called MI-FROG [19]. Multi-grid involves using smaller arrays for initial iterations before using the entire data array. In FROG and XFROG, if the measured trace is $N \times N$, the number of multiplications in GP scales as $N^2 \ln N$, so reducing the size of the array is clearly beneficial. The multi-grid approach then uses the retrieved pulse from the smaller grid as the initial guess for the larger grid. The process can be repeated and concludes after reaching the original, largest, as shown in Fig. 1.

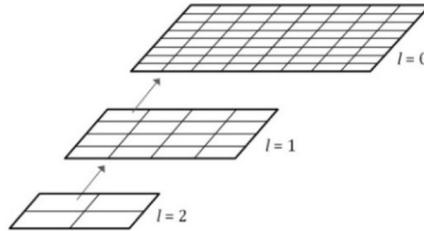


Fig. 1. Multi-grid algorithm used in GP XFROG. The parameter, l , corresponds to the reduced size array. In this work, $l = 0$ corresponds to an $N \times N$ trace; $l = 1$ corresponds to $N/2 \times N/2$, and $l = 2$ corresponds to $N/4 \times N/4$.

2. Multi-grid algorithm in GP FROG

It is straightforward to simultaneously reduce the trace delay and frequency ranges by a factor of 2. Simply use the center half of the trace in each dimension. But GP requires that the delay and frequency dimensions satisfy the discrete Fourier transform relation:

$$N = \frac{1}{\delta\tau \delta\nu} = \Delta\tau \Delta\nu \quad (2)$$

where $\delta\nu$ and $\delta\tau$ are the frequency and delay increments, respectively, and $\Delta\nu$ and $\Delta\tau$ are the frequency and delay ranges, respectively. This is reasonable because a pulse with a temporal range of $\Delta\tau$ typically has spectral structure of size $\delta\nu$. And a pulse with a spectral range of $\Delta\nu$ typically has temporal structure of size $\delta\tau$.

So obeying the discrete Fourier-transform relations between delay and frequency requires a corresponding *increase* in the trace increments by the same factor, yielding a smaller-range and coarser trace. This results in a significantly smaller, $N/4 \times N/4$, trace. It involves simply averaging every 2×2 subarray within the central region of the grid, yielding one number for every such 2×2 subarray. As a result, Siders and associates used (in reverse order) traces of sizes $N \times N$, $N/4 \times N/4$, ..., which worked very well for MI-FROG's traces.

Of course, trace *structure* will be lost in the process. And reducing the *range* of an XFROG trace—whose initial size has been correctly chosen—to smaller than $N/4 \times N/4$ can result in overly cropped traces. We have found that, typically, a properly chosen $N \times N$ trace (i.e., one with not too many zeros in its wings, that is, not overly large) can be reduced to

$N/4 \times N/4$, but no smaller, as shown in Fig. 2. Consequently, direct application of multi-grid to XFROG would yield only one smaller array, $N/4 \times N/4$.

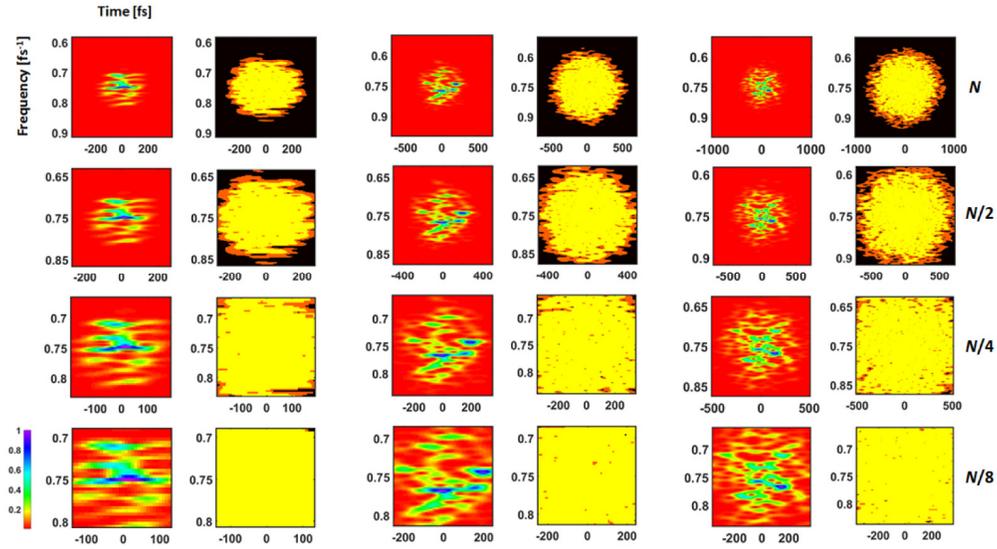


Fig. 2. For implementation of the GP algorithm, the FROG trace should be “an island in a sea of zeros.” However, due to the uncertainty principle, the trace cannot go to exactly zero on the perimeters. Here, the orange and yellow colors represent the regions where $I_{sig}(\omega, \tau)$ is greater than 10^{-4} , and 10^{-3} of the maximum value of the trace, respectively. Here each column corresponds to the trace of a different pulse. Each row represents $N \times N$, $N/2 \times N/2$, $N/4 \times N/4$, and $N/8 \times N/8$ XFROG traces for that column’s pulse that would be used for the multi-grid algorithm. Left to right: 1st column, $N = 256$ ($TBP_{rms} = 10$), 3rd column $N = 512$ ($TBP_{rms} = 30$), and 5th column $N = 1024$ ($TBP_{rms} = 55$). Notice the change in delay and frequency ranges in the smaller traces. The corresponding regions where data points are greater than 10^{-4} and 10^{-3} of the maximum of trace are shown for all the traces on the adjacent column. Thus, $N/4 \times N/4$ traces are generally acceptable, but $N/8 \times N/8$ traces are too cropped and hence usually unacceptable.

But using more than one trace-size reduction in multi-grid is desirable (as we will show), so we also extended the multi-grid method by using an additional intermediate-size array, $N/2 \times N/2$, here involving reducing the dimensions of the array by a factor of two in addition to four. As a result, the delay and frequency increments must increase, and the ranges must decrease, by $\sqrt{2}$. This is not as simple as generating the $N/4 \times N/4$ array, but it is still not difficult. We computed actual values by simple interpolation of neighboring points.

We run the algorithm first on the coarsest grid, $N/4 \times N/4$, then $N/2 \times N/2$, and finally $N \times N$. A flat-phase Gaussian pulse with $\tau_{FWHM} = 80$ fs is used as reference pulse for all the theoretical traces. We used a personal computer with Intel Core i7 3.40GHz processor, running MATLAB with the FROG algorithm kernel in C + + . We considered pulses with TBPs up to 90.

The time required for binning the 256×256 , 512×512 , and 1024×1024 arrays to two smaller grids was ~ 0.03 , 0.05 , and 0.06 s, respectively—negligible compared to the iteration times. For all the retrievals, random complex numbers were used as the initial guesses for the $N/4$ array, and the result of that retrieval was then used as the initial guess for the larger array, etc., as required for multi-grid. For runs without noise, the algorithm for size $N/2^l$ was terminated when the difference between the two successive G errors (rms differences between the traces) reached a threshold, α_l . The threshold values were found by fine-tuning their value on a predetermined coarse range based on the best performance of retrieval on a sample of pulses. For example, for $N = 512$, the algorithm on the 128×128 grid was set to terminate for $\alpha_2 = 8 \times 10^{-7}$ and for 256×256 , $\alpha_1 = 6 \times 10^{-6}$.

Finally, we used large numbers of randomly chosen traces in order to produce statistically significant results.

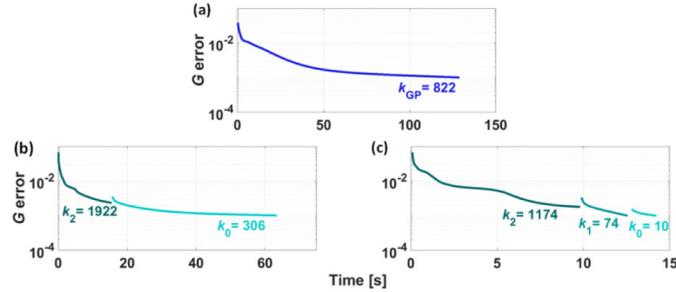


Fig. 3. Comparison of the standard GP XFROG algorithm (a), the Siders, et al., multi-grid GP XFROG algorithm (which uses only $N/4 \times N/4$ and $N \times N$ traces) (b), and our approach using $N/4 \times N/4$, $N/2 \times N/2$, and $N \times N$ traces on a 1024×1024 XFROG trace for a pulse with $TBP_{rms} = 55$. Time on the horizontal axis corresponds to real time. Note that our multi-grid GP approach is almost ten times faster than the standard GP approach for this pulse. The parameters k_l indicate the number of iterations on the $N/2^l \times N/2^l$ array. In other words, k_0 corresponds to the full $N \times N$ array, k_1 corresponds to the $N/2 \times N/2$ array, and k_2 corresponds to the coarsest $N/4 \times N/4$ traces in (b) and (c). The difference in convergence behaviors is due to the fact that many more iterations are required for a given value of l when fewer array sizes are used. For example, when the $N/2 \times N/2$ array is not used, as in (b), more iterations are required on both the coarsest and finest arrays. Note that the time per iteration is the same for all approaches and depends only on the value of l .

We find that the retrieved pulse in multi-grid GP is typically obtained using a larger number of iterations, but on much less time-costly smaller grids, which results in a much more overall time-efficient algorithm. Multi-grid always yielded faster convergence. And including the intermediate-size array, $N/2 \times N/2$, always resulted in the fastest convergence, as shown in Fig. 3. In other words, the pulse retrieval on the smaller-range coarse traces provides a good initial guess for the larger-range, finer trace and, as a result, reduces the number of required iterations on it.

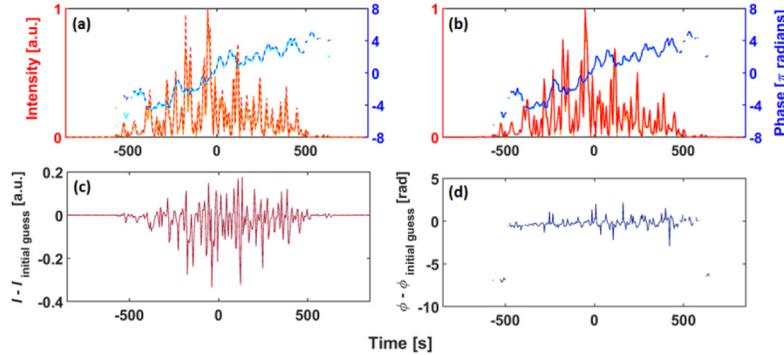


Fig. 4. (a) The temporal intensity and phase of the actual theoretical pulse are shown by orange and cyan colors, respectively. The temporal intensity and phase retrieved from the $N/2 \times N/2$ trace to be used for the initial guess for the full $N \times N$ (512×512) trace are represented by the dashed red and blue lines, respectively. Additional points have been interpolated for comparison with the full $N \times N$ array. (b) The retrieved electric field obtained from the full $N \times N$ trace with G error = 1.4×10^{-3} . The temporal intensity and phase of the retrieved pulse (indistinguishable from the actual theoretical pulse, shown in (a)) are represented by red and blue curves, respectively. (c) The difference between the intensities (dark red) of the actual field and the field retrieved from the $N/2 \times N/2$ trace and used as the initial guess for the retrieval on the fine-grid. (d) The difference between the phases (dark blue) of the actual field and the field retrieved from the $N/2 \times N/2$ trace and used as the initial guess for the retrieval on the fine-grid. Note the significant discrepancies in (c) and (d), confirming the need for at least a few iterations on the full $N \times N$ trace.

It should be noted that, at first glance, the pulse retrieved from the $N/2 \times N/2$ trace seems close to the exact pulse, as shown in Fig. 4(a). But it lacks much of the pulse's fine structure, and it occasionally adds structure, so it was important to retrieve the pulse from the complete $N \times N$ trace, as shown in Figs. 4(b)-(d). Fortunately, at most only a few iterations on the entire trace were required, so not much additional time was required for this additional accuracy.

Figure 5 represents the retrieval results for theoretical pulses with rms TBPs of 10, 30, and 55. Note the small numbers of required iterations on the full trace in multi-grid, denoted by $k_{0,N}$. Also note that k_{MG-GP} , which corresponds to the equivalent number of iterations (in terms of iteration per cost) on the fine grid is quite small compared to k_{GP} , which represents the average number of iterations in standard GP XFROG on the full array. We can see that for a 1024×1024 trace, the retrieval time is reduced by a factor of ~ 7 . Recall that the time required for the binning is negligible.

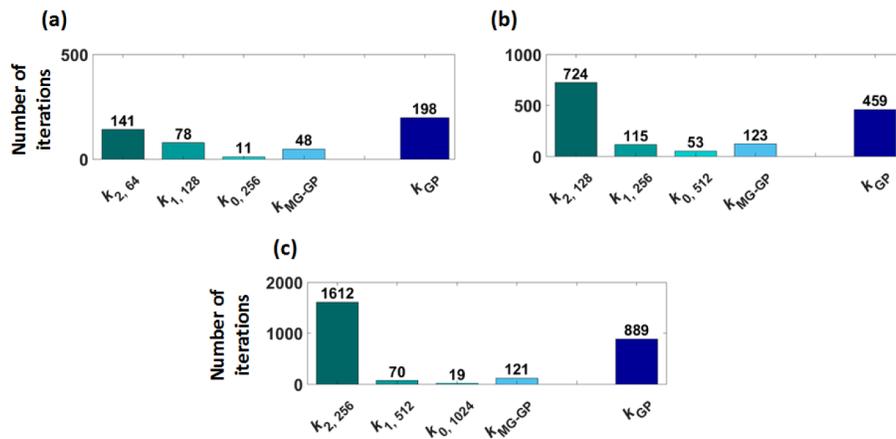


Fig. 5. Average number of iterations on a set of 50 pulses with (a) TBP = 20 and a 256×256 trace, (b) TBP = 30 and a 512×512 trace, and (c) TBP = 55 and a 1024×1024 trace. k_l corresponds to average number of iterations for the array of size reduced by the factor 2^l . k_{MG-GP} and k_{GP} correspond to equivalent number of iterations of k_l 's in terms of iterations on the fine grid, and average number of iterations required in GP XFROG, respectively.

To better simulate experimental data, we contaminated traces with 1% multiplicative plus 1% additive noise. The results were very similar, as shown in Fig. 6. For TBP > 10, multi-grid remains far preferable, providing a factor of ~ 6 improvement for retrieval on a 1024×1024 array, and ~ 7 for retrieval of TBP = 90, requiring a 2048×2048 trace.

We also find that our multi-grid approach is as robust as the GP algorithm on which it is based, with all traces yielding convergence to the correct pulse on the first initial guess, without the need for a second attempt, even in the presence of noise.

Also, recall that it has recently been shown that, in contrast to most other popular pulse-measurement techniques, FROG and XFROG are excellent indicators of pulse-shape instability when a measurement averages over many pulses [9, 10]. This is because the complete measured FROG or XFROG trace vastly overdetermines the pulse, and the presence of different pulse shapes in the pulse train distorts the trace and so essentially always yields a trace that cannot correspond to a single pulse. As a result, the FROG algorithm, which can only yield a single pulse and so assumes that the trace is due to only one pulse, not the sum of many different ones, yields a pulse with a trace that does not agree with the measured trace. Because multi-grid generalized projections utilizes the entire FROG or XFROG trace in its final iterations, it retrieves the same pulse as the standard GP algorithm and so necessarily retains FROG's ability to ascertain the pulse-shape stability of a train of pulses. Other algorithmic approaches that attempt to retrieve pulses from incomplete FROG or XFROG traces, which do not in the end use the entire trace, will not enjoy this important feature.

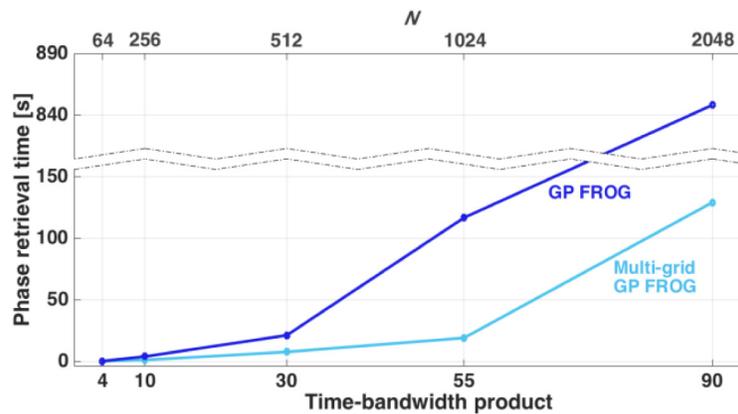


Fig. 6. The average retrieval times for sets of 50 pulses in the presence of noise. The improvement in speed is the same whether or not 1% additive and multiplicative noise is present.

3. Conclusion

For simple pulses and their corresponding small traces, FROG's and XFROG's original generalized projections algorithm converges quickly and reliably (for XFROG, typically in 0.02s—less than the time required to plot the resulting pulse). But complex pulses, such as continuum and shaped pulses, require a faster XFROG pulse-retrieval algorithm. We showed that Siders' multi-grid XFROG algorithm, modified to also include an intermediate-size trace, significantly reduces the retrieval time of the existing generalized-projections algorithm. We also considered much more complex pulses than previously considered, and we observed a factor of ~ 7 speed improvement for complex pulses with TBPs of ~ 50 to ~ 90 . This speed improvement factor is likely to increase for even larger traces. So multi-grid provides the most improvement where it is most needed.

Finally, it is worth reiterating that implementation of multi-grid is very easy, as it does not require any modification of the FROG-algorithm kernel, and instead only to the routine that calls it.

Funding

National Science Foundation (NSF) (#ECCS-1307817); the Georgia Research Alliance.

Disclosures

Rick Trebino owns a company that sells pulse-measurement devices.