Real-Time Measurement of Ultrashort Laser Pulses Using Principal Component Generalized Projections

Daniel J. Kane

***Abstract*—Frequency-resolved optical gating (FROG) is a technique to measure ultrashort laser pulses that optically constructs a spectrogram of a laser pulse. A two-dimensional (2-D) phase retrieval algorithm is used to extract the intensity and phase of a pulse from its spectrogram. We have improved a recently presented principal component generalized projections algorithm (PCGPA) making it easier to implement and fast enough to allow real-time inversion of FROG spectrograms. A femtosecond oscilloscope, based on second-harmonic generation (SHG) FROG, that displays the intensity and phase of ultrashort laser pulses in real time utilizing the improved PCGP phase retrieval algorithm is presented.**

***Index Terms*—Optical pulse measurements, optical signal processing, signal processing, ultrafast optics.**

I. Introduction

FREQUENCY-RESOLVED optical gating (FROG) determines the intensity and phase of ultrashort laser pulses by optically constructing a spectrogram of a pulse in order to characterize it [1]–[18]. The spectrogram, or FROG trace, is constructed by combining a gate pulse and the pulse to be measured in an instantaneously (or nearly so [15]) responding nonlinear optical medium. The resulting signal is a time slice of the pulse which is spectrally resolved. As the gate is scanned across the pulse in time, all of the time-frequency information about the pulse is recorded. The gate may either be a known function of the pulse to be measured (FROG) as in the case of a spectrally resolved autocorrelation [1]–[9], [13], [14], or the gate may be an unknown, unrelated laser pulse (TREEFROG or blind-FROG) [10], [11]. An iterative two-dimensional (2-D) phase retrieval algorithm is used to determine the spectrogram's phase and hence, the original pulse [8]–[10].

While FROG is experimentally simple, the phase retrieval computation can be slow. Immediate information about the pulse must be gleaned from its FROG trace; quantitative information must wait for results from the inversion algorithm which may require over a minute. Real-time optimization of system performance (that is, adjustments of pulse duration, shape and chirp) requiring quantitative information become difficult and tedious.

The best FROG inversion algorithms arrive at the optimal solution quickly, either by iterating quickly, by converging in few iterations, or a combination of both without stagnating at nonoptimal solutions. The original FROG inversion algorithm, commonly referred to as the vanilla algorithm [7], [8], is simple and iterated quickly, but tends to stagnate, giving erroneous results, especially for geometries that use a complex gate function such as second-harmonic generation (SHG) and self-diffraction. An improved algorithm, incorporating several different algorithms including brute-force minimization, was developed to alleviate stagnation problems at the expense of both iteration speed and convergence time [8]. A significant advance in both speed and stability was made by the addition of a numerical method called generalized projections [9], [13]; the slowest part of this implementation is a step to determine the next guess for the electric field which minimizes the error (distance) between the FROG electric field and the next guess for the electric field.

Recently, a new generalized projections algorithm has been developed called principal component generalized projects algorithm (PCGPA) that eschews the need for minimization [10]. Based on the idea that a FROG trace can be constructed from an outer product of two vectors representing the pulse and gate, the new algorithm reduces the construction of a new guess for the pulse and gate to the calculation of two eigenvectors. We now report an additional improvement to PCGPA that overcomes two problems with previous implementations. First, the original implementation of PCGPA requires a time consuming step called a singular value decomposition (SVD) [10]. Now, we replace the slow SVD step with a fast matrix-vector multiply producing a real-time FROG inversion algorithm. The second problem is that PCGPA is inherently a blind-FROG or TREEFROG [10], [11] algorithm that obtains the pulse and gate independently. When used to invert FROG traces where the gate is constructed from the pulse itself, ambiguities can degrade the accuracy of the retrieved pulse [10]. Symmetrizing the outer product removes the ambiguities, restoring the accuracy of the retrieved pulse. Furthermore, we demonstrate the improvements by constructing a femtosecond oscilloscope based on SHG FROG [13], [14] that displays the intensity and phase (albeit a time-reversal ambiguity inherent in SHG FROG [1], [13], [14]) in real time-greater than 2 Hz.

II. Principal Component Generalized Projections

As a review, the SVD-based PCGP algorithm is depicted in Fig. 1 [10]. Initial guesses of random noise modulated by a broad Gaussian for the pulse and gate are used to construct the first outer product matrix. A one-to-one transformation via permutations converts the outer product into a time-domain

Manuscript received September 30, 1997; revised February 12, 1998. This work was supported by the National Science Foundation under Grant DMI-9 661 596 and by the U.S. Air Force under Contract F33615-96-C-2632.

The author is with Southwest Sciences, Inc., Santa Fe, NM 87505 USA.

Publisher Item Identifier S 1077-260X(98)03773-3.

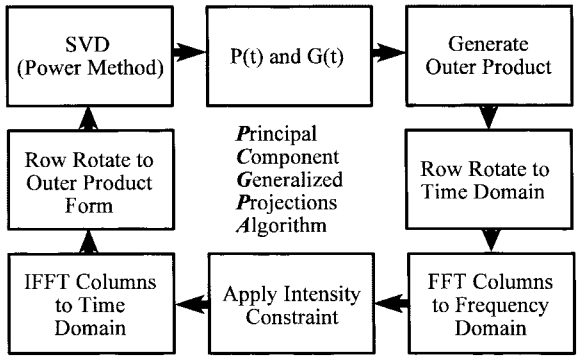

Fig. 1. Schematic of the PCGP algorithm. By replacing the SVD with matrix-vector multiplies using the power method, the algorithm can be made to run significantly faster.

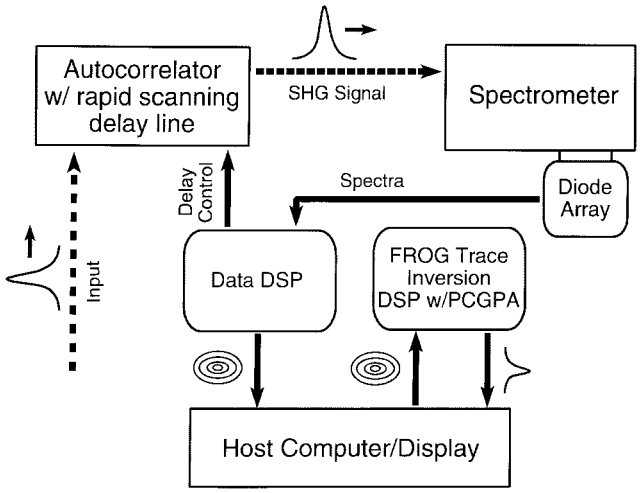

Fig. 2. Schematic of the SHG FROG device. The device used two DSP cards based on floating point DSP processors. One was used for data acquisition and the other was used for the inversion engine. The data acquisition DSP also controls the delay line in the autocorrelator and formats the data (spectrogram) for the inversion engine. The host computer moves the data from the data acquisition board to the inversion engine DSP programmed to run the SHG FROG PCGP algorithm. The algorithm ran at about 20 iterations/s for a 64×64 FROG trace, and for a 32×32 FROG trace, it ran at about 60 iterations/s. The host computer is also used for user I/O, displaying the FROG trace, and the inverted pulse. With proper synchronization, this device would also be compatible with high repetition rate, amplified, ultrafast lasers.

FROG trace of the initial guess [10]. The columns are Fourier transformed and an intensity constraint (the experimental FROG trace) is applied. The result is transformed back to the time domain, and an inverse of the outer-product-to-time-domain transform is applied. The outer product "form" matrix is decomposed into a superposition of weighted outer products by an SVD. The vector pair (outer product) with the largest corresponding weight is the best rank one approximation of the outer product form in the least squares sense; hence, this guess is a projection [9], [13], [19], [20] and is used for the next iteration.

While simple to implement, an SVD is computationally intensive and wasteful; it generates $N-1$ unnecessary vector pairs. Fortunately, the principal vector pair may be found directly with much less computation, reducing the SVD step to simple matrix-vector multiplies.

First, the outer product form matrix is an $N \times N$ matrix, O. There are two sets of N orthonormal eigenvectors P_i and G_i such that

$$\begin{aligned} OO^T P_i &= \lambda_i P_i \\ O^T O G_i &= \lambda_i G_i \end{aligned} \tag{1}$$

where λ_i are the eigenvalues, or "weights," and the superscript T is the transpose operator. O may be constructed by

$$O = \sum_{i=1}^{N} \sqrt{\lambda_i} P_i G_i^T \tag{2}$$

where λ_i, P_i, and G_i are provided by the SVD, but we only need the P_i and G_i corresponding to the largest $|\lambda_i|$, or the principal eigenvectors. Suppose we multiply an arbitrary nonzero vector x_0 by OO^T. Then

$$OO^T x_0 = \sum_{i=1}^{N} \kappa_i \lambda_i P_i \tag{3}$$

where P_i are the eigenvectors of OO^T, λ_i the eigenvalues, and κ_i a set of constants. Because $OO^T \kappa_i \lambda_i P_i = \kappa_i \lambda_i^2 P_i$, multiplying by $(OO^T)^{p-1}$ gives

$$(OO^T)^p x_0 = \sum_{i=1}^{N} \kappa_i \lambda_i^p P_i. \tag{4}$$

Thus, as p becomes large, the largest eigenvalue, λ_l, dominates the sum so that $(OO^T)^p x_0 \sim \kappa_l \lambda_l^p P_l$. This method is called the *power method* [21].[1] After a few iterations, a very close approximation to the principal eigenvector (the eigenvector with the greatest eigenvalue) is obtained. Consequently, the next guess for the pulse can be obtained by multiplying the previous guess for the pulse by OO^T. The next guess for the gate can be obtained by multiplying the previous guess for the gate by $O^T O$ (for polarization gating, the absolute value of the result for the gate is taken). While better approximations for the eigenvectors may be obtained by using these operators several times per iteration, once per iteration is adequate in practice. While requiring a few more iterations to converge, overall, the power method implementation gives about a factor of 20 improvement in convergence speed over the SVD implementation.

PCGPA is inherently a blind-FROG [10] or TREEFROG [10], [11] algorithm that finds the probe and gate independently without any assumptions about relationships between them. As a result, it is prone to some ambiguities involving the width of the gate and probe pulses [10]. For example, a very slight change in the width of the probe may be compensated for by the algorithm by a slight change in the width of the gate without changing the rms error significantly. In FROG, as opposed to blind-FROG, because of the *a priori* knowledge of the relationship between the probe and gate, the width ambiguity is not a problem. These ambiguities may be resolved when using a blind-FROG inversion algorithm such as PCGPA by the addition of a spectral constraint on either the probe or

[1] The power method depends on the outer product form matrix having only one dominant eigenvalue. In practice, this is almost always true. If not, the FROG trace is most likely pathological, resulting from the superposition of two or more traces, or from severe distortions.

the gate [10]. In PCGPA, the spectral constraint is applied after the intensity constraint is applied and just before the next guess is computed; hence, the FROG trace is in the outer product form. When in this form, each column is ideally a constant (one of the elements of the gate) multiplied by the probe field. Thus, each column is Fourier transformed and the magnitude is replaced by the square root of the measured pulse spectrum. (To insure the gate field is preserved, the area of the intensities before and after the spectral constraint is applied are kept equal. To prevent artifacts from appearing in the wings of the FROG trace, the spectral constraint may only be applied to portions of the trace that has an integral above some predetermined level.) The columns are then inverse Fourier transformed back to the time domain before applying the SVD (or power method).

While the use of spectral constraints can facilitate inversion of FROG spectrograms, for practical applications, it is desirable to reduce the complexity of the device; in other words, we would rather not be required to obtain a spectrum of the pulse in addition to its spectrogram. Thus, improvements in the PCGPA algorithm to permit the inversion of FROG traces rather than blind-FROG traces are advantageous. The conversion of PCGPA to a FROG algorithm may be accomplished as follows. Assume we have a nonlinear medium that produces a gate from the input pulse via a function we will call Γ; its inverse is Γ^{-1}. Rather than using only the outer product of E_{probe} and E_{gate} to produce the next time domain FROG trace, the sum of the outer products of $E^i_{\text{probe}}E^j_{\text{gate}}$ and $\Gamma^{-1}(E_{\text{gate}})^i\Gamma(E_{\text{probe}})^j$ is used so that the outer product on the next iteration is given by

$$O^{ij}_k = \text{probe}^i_k \text{gate}^j_k + \Gamma^{-1}(\text{gate}_k)^i\Gamma(\text{probe}_k)^j \qquad (5)$$

where O_k is the symmetrized outer product for the kth iteration. In the case of SHG FROG, the function that produces the gate from the probe and its inverse is trivial, reducing the inverse to the conjugate of the transpose.

This type of constructed FROG algorithm works very well for SHG FROG. Table I compares the PCGP-based SHG FROG algorithm to the commercially available Femtosoft SHG FROG algorithm, and to the "Basic FROG" or "Vanilla" algorithm. The three algorithms were tested to failure by three synthetically constructed test sets (See Appendix). From the test results, it can be determined that the generalized projections-based algorithms are clearly superior to the "Vanilla" algorithm. In the first test, the random noise test, which determines the overall robustness of the algorithms, the SHG PCGP algorithm performed best. In the other two tests, the PCGP SHG algorithm compares favorably to the Femtosoft algorithm. The tests were conducted as follows. A pulse was constructed randomly that complied to a fixed set of criteria (See Appendix). A spectrogram was constructed and each algorithm was allowed to run for 100 iterations. Each such test was repeated for at least 25 different pulses. The first number indicates percentage of test pulses that converged to an error of 0.2% or less which is approximately the lowest experimental error expected and is, therefore, experimentally the most useful number. The number in parenthesis is the percentage of test pulses that converged to an error of 10^{-4} or less, which is the nominal value for strict convergence.

TABLE I
COMPARISON OF SHG FROG INVERSION ALGORITHMS

Algorithm Type	Random Noise Test: Percent Convergence	Random Chirp Test: Percent Convergence	Multi-pulse Test: Percent Convergence
SHG Vanilla	15 (4)	45 (32)	64 (3)
Femtosoft SHG FROG	56 (48)	80 (52)	92 (0)
Fast SHG PCGPA	78 (77)	79 (70)	95 (10)

The percentages given are for percent of pulses retrieved with an rms FROG trace error of less than 2×10^{-3} in 100 iterations. This was deemed to be the lowest experimental error that can in practice be achieved and more aptly defines the usefulness of the algorithm. Percentages given in parenthesis are for strict convergence with an rms error of less than 10^{-4} in 100 iterations. The ultimate convergence rates (allowing the algorithm to continue past 100 iterations until stagnation) for the Femtosoft SHG FROG inversion algorithm were 60%, 80%, 32% for the random noise test, the random chirp test, and the multipulse test, respectively. The ultimate convergence rates of the SHG FROG PCGP algorithm was not determined.

Using PCGPA and the inverse method (only for SHG FROG is the outer product truly symmetric) also works well for polarization-gate (PG) FROG on the synthetic test sets, converging to experimental error for 90% of the test pulses, but it has not been tested with experimental data. In the PG FROG, the inverse of the gate function does not exist. As a result, a pseudo-inverse must be constructed from the square root of the gate intensity and the phase of the pulse. However, the square root can cause small fluctuations in the wings of the gate producing artifacts in the next guess for the pulse which causes instabilities in the algorithm. This can be remedied by applying the square root only in portions of the gate where it is well defined. Where the gate is not well defined, the intensity (and phase) of the pulse is used. To increase the robustness of the PG algorithm, the symmetrization constraint is applied on alternate iterations. The pseudo inverse method does not appear to work well for self-diffraction (SD) FROG, however.

III. EXPERIMENTAL—THE FEMTOSECOND OSCILLOSCOPE

A multishot SHG FROG device was built that seamlessly integrated data acquisition and the inversion algorithm by using commercially available digital signal processing (DSP) boards (Fig. 2). This device successfully demonstrated the inversion of experimental FROG traces in real-time and could display the inverted pulses (from a 64×64 FROG trace) at a rate of 1.25 Hz, or one every 0.8 s, 2.3-Hz inversion rates were possible for a 32×32 array.

Example data obtained using the femtosecond oscilloscope are shown in Fig. 3. Also show in Fig. 3 are the retrieved pulse, the retrieved phase, and an example of the performance of the DSP/PCGPA combination. After only one second of computational time, the algorithm ran for 20 iterations on the 64×64 FROG trace, converging to a FROG trace error [7] of less than 0.5%. The algorithm was allowed to continue for another $\sim$90 000 iterations, but this did not significantly change the retrieved pulse and phase.

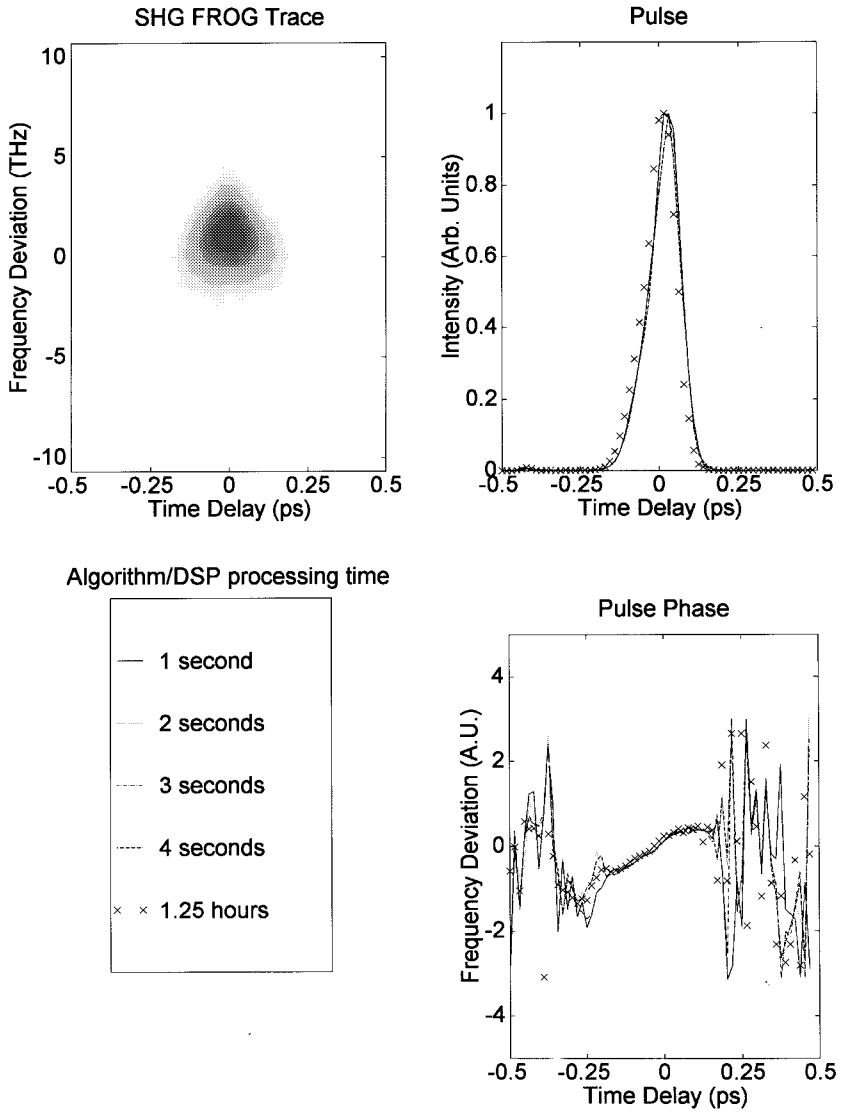

Fig. 3. FROG trace and retrieved pulse obtained from the femtosecond oscilloscope shown in Fig. 2. The FROG trace is in the upper left corner of the figure. In the upper right-hand corner is the retrieved pulse. This pulse was retrieved using the PCGP algorithm on the inversion engine DSP. On this DSP board, the algorithm ran at approximately 20 iterations/s. In only one second, the algorithm converged to an error of less than 0.5%. Also, the algorithm is stable; the results did not change significantly even after 90 000 iterations, which is an important criteria for a real time inversion algorithm.

To further test the femtosecond oscilloscope, a pulse stretcher-compressor was built to vary the pulse dispersion independent from the Ti:sapphire oscillator (Fig. 4). The basic configuration of the zero dispersion stretcher/compressor is the same as that used by Chilla and Martinez [22]. At the input of the stretcher/compressor was a 1200 g/mm grating. The first-order dispersed light was sent through a 175-mm focal length lens approximately one focal length away. A mirror, also one focal length away, reflected the dispersed light back through the lens onto the grating. By translating the lens, dispersion in the beam could be changed enough to more than triple the pulse width. The femtosecond oscilloscope could easily track these changes. Also, portions of the spectrum could be blocked to shape the pulse before being sent to the FROG device (Fig. 5).

An SHG FROG device, or spectrally resolved autocorrelator [13], [14] was used in the femtosecond oscilloscope described in this work. The input beam is split into two identical beams by a beam splitter. One beam is sent into a manual delay line used to fine tune the delay between the two beams so that the pulse and gate are equivalent for proper operation of the PCGP algorithm. The other beam is sent into a fast scanning delay line. This delay line is a 0.5-in diameter retroreflector controlled by a General Scanning LT 1000 Z (linear) scanner allowing the delay to be controlled by a voltage ($\sim$2 mm delay/V). The resulting beams were about 8 mm apart and focused by a 250-mm fl lens into a 200-μm-thick BBO crystal. The spectrum of the second harmonic is measured via a spectrograph and an EG&G Reticon diode array controlled by the EG&G demonstration board. The signal from the diode array electronics was sent into an SRS 560 low-noise differential amplifier before being digitized by the 16-b A/D converters on the data collection DSP board. The pixel clock on the diode array controller was set to 100 kHz (exposure of 5 ms), the maximum sample rate of the A/D's. After the diode array was read, the translation stage was set to

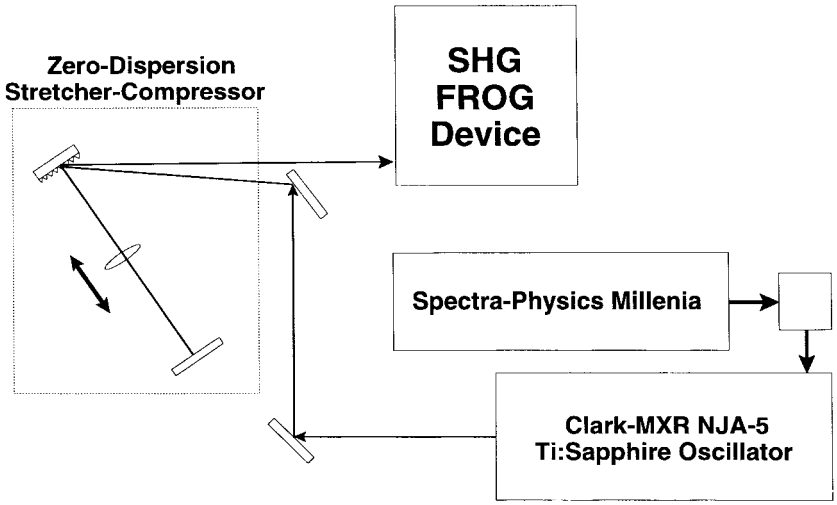

Fig. 4. A schematic of the experimental arrangement used. The zero-dispersion stretcher-compressor allowed pulse parameters to be varied independent of the oscillator. The femtosecond oscilloscope could easily track changes made in the dispersion of the pulse.

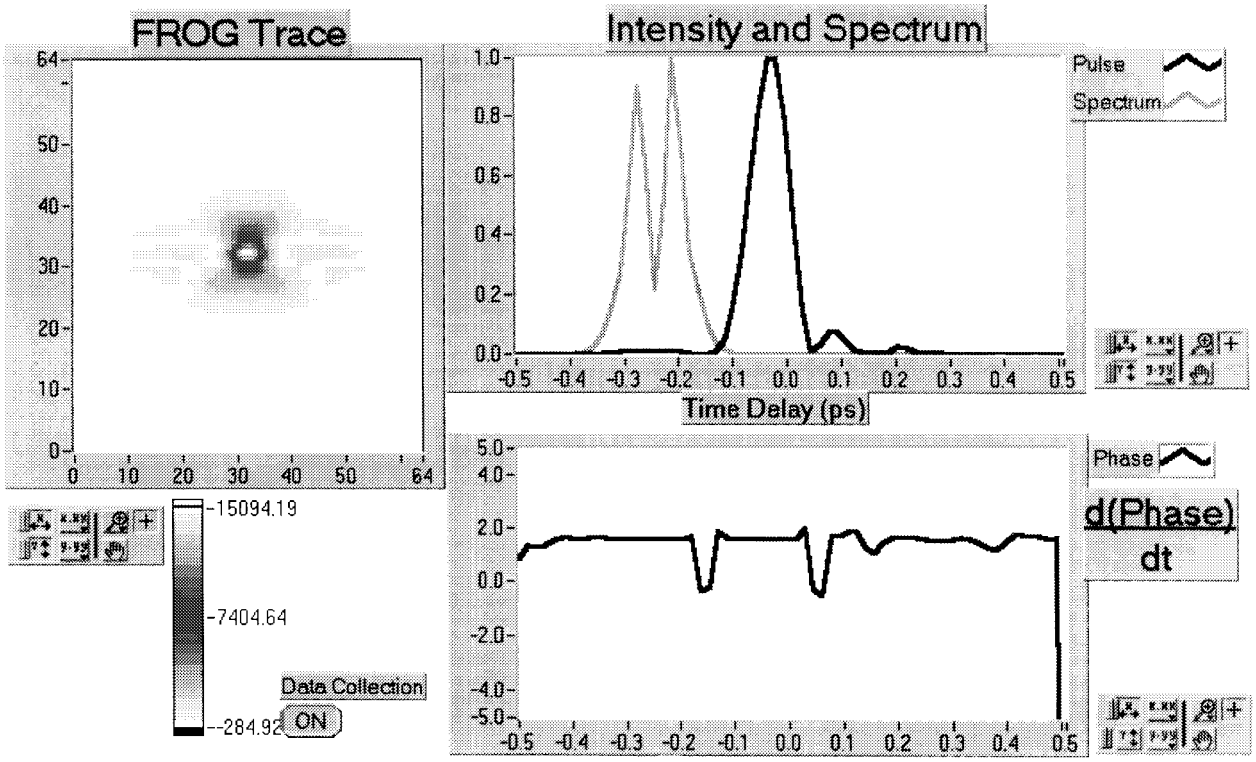

Fig. 5. Front panel display of the femtosecond oscilloscope. Both the pulse intensity and spectrum are shown in the upper right-hand corner of the display. The pulse was obtained by placing a wire in the stretcher-compressor to block out some frequencies. The ringing of the pulse is clearly visible. The first derivative of the phase (frequency deviation) is displayed so that the variable arbitrary offset does not change the scale of the plot between refreshes; thus, more subtle changes in the phase can be observed.

the next delay via a D/A on the DSP board. Sixty four spectra were obtained for the 64×64 FROG trace, 32 for the 32×32 FROG trace. Only every other spectra was sampled, resulting in a throughput of 98 spectra/s.

The data collection DSP board also prepared the raw data for input into the algorithm by resampling the signal vector from the 512-element diode array down to 64 pixels using a 15-element FIR digital filter. The coefficients were chosen to remove all frequencies higher than Nyquist for the resampled vector. After filtering, the background from electronics offset and scattered light is subtracted.

A host computer (166 MHz Pentium) controls both DSP boards which are each based on a single Texas Instruments TMS320C32 floating point DSP (Fig. 2). A dynamically linked library (DLL) was used for host-DSP communication and DSP control allowing the host program to be written in a high level language such as MATLAB or LabVIEW. The host program polls the data acquisition DSP board for a new

spectrogram. When ready, the host reads the spectrogram and frees the board to read another spectrogram. The host then reads the new pulse and gate from the inversion engine DSP board running the SHG FROG PCGP algorithm. The new spectrogram is then sent to the inversion engine board. The initial guess used by the algorithm in the inversion DSP for the new spectrogram is the pulse retrieved from the previous spectrogram. The process is repeated indefinitely.

The host program is also responsible for the user interface and display. Since all of the DSP control can be confined to DLL's, the user interface and host control program was written in LabVIEW. The "front panel" for the test program is shown in Fig. 5. The FROG trace is shown in the left-hand portion of the figure. The plot in the right-hand corner shows the intensity of the measured pulse and its calculated spectrum. The displayed pulse was formed by placing a wire in the stretcher-compressor to block out the center of the spectrum. Ringing of the pulse intensity caused by the loss of the center frequencies is clearly visible to the right of the pulse. The plot just below the intensity plot shows the time derivative of the phase of the pulse (time domain).

IV. Conclusion

A significant increase in speed is obtained when the power method is applied to PCGPA without any noticeable performance decrease [10]. For SHG FROG, the PCGP algorithm operates about two to four times faster than the current generalized projections algorithm while being as robust as the commercially available compound algorithm [19]. A femtosecond oscilloscope was constructed that obtained SHG FROG traces, and used the improved PCGP algorithm [9], [13], [19] to invert them. The display update of the FROG trace, and the intensity and phase of the pulse, was real time with an update of 0.8 s or 1.25 Hz for a 64×64 array (0.43 s or 2.3 Hz for a 32×32 array).

Appendix
Construction of Test Sets

A. General Issues

The idea behind these test sets is to develop a way to compare the performance of two different algorithms statistically. Three different test sets were constructed in order to model the performance of the algorithms for different pulses. The three test sets are: 1) Filtered random noise; 2) random amounts of well behaved phase variations; and 3) multiple pulses with slight phases distortions.

B. Random Noise Test Set

This test is intended to test the general robustness of the algorithm. The pulses that made up the random noise test were made by first making a complex vector 64 points long by producing a real and an imaginary vector each with values varying between -1 and 1. These vectors were then multiplied by a Gaussian with a FWHM of 10.5 pixels. The result was filtered to remove all frequencies above 30% of Nyquist to prevent aliasing when the FROG trace was constructed.

C. Random Phase Test Set

This test is intended to simulate pulses obtained from a chirped pulse amplifier system. A Gaussian with a FWHM of 9.5 pixels with randomly varying quantities of well-defined phase distortions. The phase distortions in the time domain (and the normalized width) were linear chirp ($0.025/\text{pixel}^2$), cubic chirp ($0.0025/\text{pixel}^3$), quartic chirp ($0.0001/\text{pixel}^4$) and self-phase modulation (2.5 rad). The phase distortions in the frequency domain (and the normalized width) were cubic chirp ($0.000\,52/\text{pixel}^3$) and quartic chirp ($8.4 \times 10^{-6}/\text{pixel}^4$).

D. Multiple Pulse Test Set

The purpose of this test was to determine the algorithm's ability to be used in pulse shaping experiments and data communication experiments. A pulse made up of no less than two subpulses and no more than five subpulses was constructed. Slight time-domain phase distortions were also introduced consisting of random amounts of linear (normalized width of $0.005/\text{pixel}^2$) and cubic chirp ($0.0005/\text{pixel}^3$ normalized width) as well as self-phase modulation (0.25-rad normalized width).

References

[1] D. J. Kane and R. Trebino, "Characterization of arbitrary femtosecond pulses using frequency-resolved optical gating," *IEEE J. Quantum Electron.*, vol. 29, pp. 571–579, 1993.

[2] ——, "Single-shot measurement of the intensity and phase of an arbitrary ultrashort laser pulse by using frequency-resolved optical gating," *Opt. Lett.*, vol. 18, pp. 823–825, 1993.

[3] 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.*, vol. 19, pp. 1061–1063, 1994.

[4] B. Kohler, V. V. Yakovlev, K. R. Wilson, J. Squier, K. W. DeLong, and R. Trebino, "Phase and intensity characterization of femtosecond pulses from a chirped-pulse amplifier by frequency-resolved optical gating," *Opt. Lett.*, vol. 20, pp. 483–485, 1995.

[5] S. Backus, J. Peatross, Z. Zeek, A. Rundquist, G. Taft, M. M. Murnane, and H. C. Kapteyn, "16-fs, 1-μJ ultraviolet pulses generated by third-harmonic conversion in air," *Opt. Lett.*, vol. 21, pp. 665–667, 1996.

[6] C. W. Siders, A. J. Taylor, and M. C. Downer, "Multipulse interferometric frequency-resolved optical gating; real-time phase sensitive imaging of ultrafast dynamics," *Opt. Lett.*, vol. 22, pp. 624–626, 1997.

[7] 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. Amer. A*, vol. 10, pp. 1101–1111, 1993.

[8] K. W. DeLong and R. Trebino, "Improved ultrashort pulse-retrieval algorithm for frequency-resolved optical gating," *J. Opt. Soc. Amer. A*, vol. 11, pp. 2429–2437, 1994.

[9] K. W. DeLong, D. N. Fittinghoff, R. Trebino, B. Kohler, and K. Wilson, "Pulse retrival in frequency-resolved optical gating based on the method of generalized projections," *Opt. Lett.*, vol. 19, pp. 2152–2154, 1994.

[10] D. J. Kane, G. Rodriguez, A. J. Taylor, and T. S. Clement, "Simultaneous measurement of two ultrashort pulses from a single spectrogram in a single shot," *J. Opt. Soc. Amer. B*, vol. 14, pp. 935–943, 1997.

[11] K. W. DeLong, R. Trebino, and W. E. White, "Simultaneous recovery of two ultrashort laser pulses from a single spectrogram," *J. Opt. Soc. Amer. B*, vol. 12, pp. 2463–2466, 1995.

[12] V. Wong and I. A. Walmsley, "Linear filter analysis methods for ultrashort-pulse-shape measurements," *J. Opt. Soc. Amer. B*, vol. 12, pp. 1491–1499, 1995.

[13] K. W. DeLong, R. Trebino, J. Hunter, and W. E. White, "Frequency-resolved optical gating with the use of second-harmonic generation," *J. Opt. Soc. Amer. B*, vol. 11, pp. 2206–2215, 1994.

[14] J. Paye, M. Ramaswamy, J. G. Fujimoto, and E. P. Ippen, "Measurement of the amplitude and phase of ultrashort light pulses from spectrally resolved autocorrelation," *Opt. Lett.*, vol. 18, pp. 1946–1948, 1993.

[15] 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.*, vol. 20, pp. 486–488, 1995.
[16] A. J. Taylor, G. Rodriguez, and T. S. Clement, "Determination of n_2 by direct measurement of the optical phase," *Opt. Lett.*, vol. 21, pp. 1812–1814, 1996.
[17] P. R. Bolton, A. B. Bullock, C. D. Decker, M. D. Feit, A. J. P. Megofna, P. E. Young, and D. N. Fittingholf, "Propagation of intense, ultraviolet laser pulses through metal vapor: Refraction-limited behavior for single pulses," *J. Opt. Soc. Amer. B*, vol. 13, pp. 336–346, 1996.
[18] J. N. Sweetser, D. N. Fittingholf, and R. Trebino, "Transient-grating frequency-resolved optical gating," *Opt. Lett.*, vol. 22, pp. 519–521, 1997.
[19] R. Trebino, K. W. DeLong, D. N. Fittinghoff, J. Sweetser, M. A. Krumbügel, and D. J. Kane, "Measuring ultrashort laser pulses in the time-frequency domain using frequency-resolved optical gating," *Rev. Scientific Instruments*, vol. 68, pp. 3277–3295, 1997.
[20] A. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1992, pp. 176–180.
[21] H. Anton, *Elementary Linear Algebra*, 2nd ed. New York: Wiley, 1997, pp. 303–310.
[22] J. L. A. Chilla and O. E. Martinez, "Direct determination of the amplitude and the phase of femtosecond light pulses," *Opt. Lett.*, vol. 16, pp. 39–41, 1991.

Daniel J. Kane was born in Wilmington, DE, in 1960. He received his B.S. degree in physics from Montana State University, Bozeman, MT, in 1983, and the M.S. and Ph.D. degrees in physics from the University of Illinois, Urbana-Champaign, in 1985 and 1989, respectively. His dissertation research involved Rydberg spectroscopy of rare gas dimers.

From 1989 to 1992, he was a Post-Doctoral Fellow in Group CLS-4 at Los Alamos National Laboratories studying nonlinear optics and ultrafast laser pulse measurement. In 1992, he joined Southwest Sciences, Inc., where his is currently employed. His interests are nonlinear optics, ultrafast pulse measurement, diode laser spectroscopy, and turbulent combustion.

Dr. Kane is a member of the American Physical Society, the Optical Society of American, and Sigma Xi.