SANDIA REPORT

SAND2009-3236
Unlimited Release
Printed June 2009

LIVA: A Data Reduction Program for Line-imaging ORVIS Measurements

Tommy Ao

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94-AL85000.

Approved for public release; further dissemination unlimited.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone:	(865) 576-8401
Facsimile:	(865) 576-5728
E-Mail:	reports@adonis.osti.gov
Online ordering:	http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone:	(800) 553-6847
Facsimile:	(703) 605-6900
E-Mail:	orders@ntis.fedworld.gov
Online ordering:	http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2009-3236
Unlimited Release
Printed June 2009

LIVA: A Data Reduction Program for Line-imaging ORVIS Measurements

Tommy Ao
Dynamic Material Properties

Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185

Abstract

LIVA (Line-Imaging Velocimetry Analysis) is a program for reducing data of a line-imaging optically recording velocity interferometer system (ORVIS) diagnostic. LIVA uses the Fourier transform method to extract phase information from recorded streak camera images. The extracted phase shift is used to infer target velocity as a function of space and time. The program can be run in any current version of MATLAB (2008a or later) or as a Windows XP executable.

Acknowledgments

The author would like to thank Wayne Trott for the valuable discussions on line-imaging ORVIS. Dan Dolan provided MATLAB support and compiled the program into a Windows executable. The author gratefully acknowledges the efforts of the DICE facility technicians for obtaining the experimental data used in the analysis examples: Randy Hickman and Jesse Lynch for operating the Veloce pulsed power generator, and Sheri Payne and Richard Hacking for fielding the line-imaging ORVIS diagnostic.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The line-imaging ORVIS (optically recording velocity interferometer system) diagnostic enables the measurement of spatially resolved velocity histories of materials in dynamic compression experiments. The basis of line-imaging ORVIS is traced back to the velocity interferometer system for any reflector (VISAR) originally developed by Barker and Hollenbach [1] in early 1970s. For decades, VISAR has been the primary time-resolved velocity diagnostic for dynamic compression experiments. The essential operation of the VISAR is to collect reflected light from a moving object and pass it through an interferometer system. The interferometer combines the input signal with a delayed version of itself to form an output signal that is sensed with fast optical detectors. The velocity of a single point on the moving object is inferred from the fringe shift of the measured signal.

To improve the time resolution of the VISAR, in 1983 Bloomquist and Sheffield [2] replaced the nanosecond risetime photomultiplier used for recording the fringe signal with a picosecond electronic streak camera, which they called an ORVIS. The line-imaging mode of the ORVIS is an adaptation of that single-point ORVIS and was demonstrated in 1995 by Baumung *et al.* [3]. Currently, line-imaging ORVIS has been implemented on various dynamic loading platforms such as gas guns [4], lasers [5], and pulsed power generators [6, 7, 8].

Data reduction of the line-imaging ORVIS interferogram involves extraction of the phase as a function of space and time. Various methods have been proposed for extracting the phase information such as fringe center tracking [9, 10], push-pull quadrature reduction [3, 4], and a Fourier transform method [11, 12]. At Sandia National Laboratories, the current procedure for reducing line-imaging ORVIS data involves several steps (image processing, velocity extraction, plotting) and multiple software programs (ImaqVision, VISAR2K, HiQ). That analysis package needs replacement because it requires obtaining software programs that are either expensive or no longer commercially available, and the original designers are no longer available to support it.

This report describes the new LIVA (Line-Imaging Velocimetry Analysis) program that has been developed for efficient and robust analysis of line-imaging ORVIS data. In order to be readily available to users within Sandia's dynamic compression community, the program was also designed to be fairly easy to implement and use.

1.1 Program summary

The LIVA program is completely self-contained and employs a simple graphical user interface (GUI) to guide the user through the analysis process. It is written in MATLAB and can operate on any platform (Mac OS X, Windows XP/Vista, and Linux) with MATLAB version 2008a or later. For non-MATLAB users, a compiled executable is also available for Windows XP. The LIVA program can be obtained by contacting Tommy Ao (tao@sandia.gov).

1.2 Overview of report

Chapter 2 gives a brief theoretical background of line-imaging ORVIS measurements and covers some aspects of the analysis. Chapter 3 gives specific details about installing and using the LIVA program. In chapter 4, examples of typical line-imaging ORVIS measurements are analyzed step-by-step with LIVA. The program's capabilities and future releases is summarized in chapter 5.

Chapter 2

Theoretical background

This chapter provides some theoretical concepts behind the analysis technique of the LIVA program. First, an overview of the operation of the line-imaging ORVIS diagnostic is given. Although several data reduction techniques have been applied to line-imaging ORVIS data, only the Fourier transform method [11, 12] used in the LIVA program will be presented here.

2.1 Line-imaging ORVIS

Line-imaging ORVIS is based on the operating principles of the VISAR diagnostic which has been described in detail elsewhere [1, 13]. Instead of observing the center of a circular interference pattern like in VISAR, the two coherent beams of monochromatic light interfere to produce a parallel set of bright lines at a recombination plane. This is accomplished by rotating the delay leg mirror M2 from its conventional orientation by an angle $\alpha/2$, as shown in Fig. 2.1. The frequency of the reflected light from the target is Doppler shifted as the surface moves, which results in a continuously changing wavelength during the acceleration of the surface. Because of the time delay difference in the two legs of the interferometer, the wavelengths of the two recombined beams are slightly different and changing during surface acceration. This produces an apparent motion of the fringe pattern in a direction perpendicular to the lines of the fringe pattern (see Fig. 2.2).

The velocity of the reflector is linearly proportional to the vertical position of the fringe pattern and the direction of accleration is determined by the direction of the fringe motion. The velocity history is given as

$$v(t-\tau/2) = \frac{\lambda_0 x(t)}{2\tau d(1+\Delta\nu/\nu_0)(1+\delta)}, \tag{2.1}$$

where τ is the temporal difference between the two legs of the interferometer, λ_0 is the laser wavelength, $(1+\delta)$ is the correction term for the frequency dispersion of the etalon, $(1+\Delta\nu/\nu_0)$ is the correction for the change in index of refraction of the window material, and d is the fringe spacing on the streak camera slit.

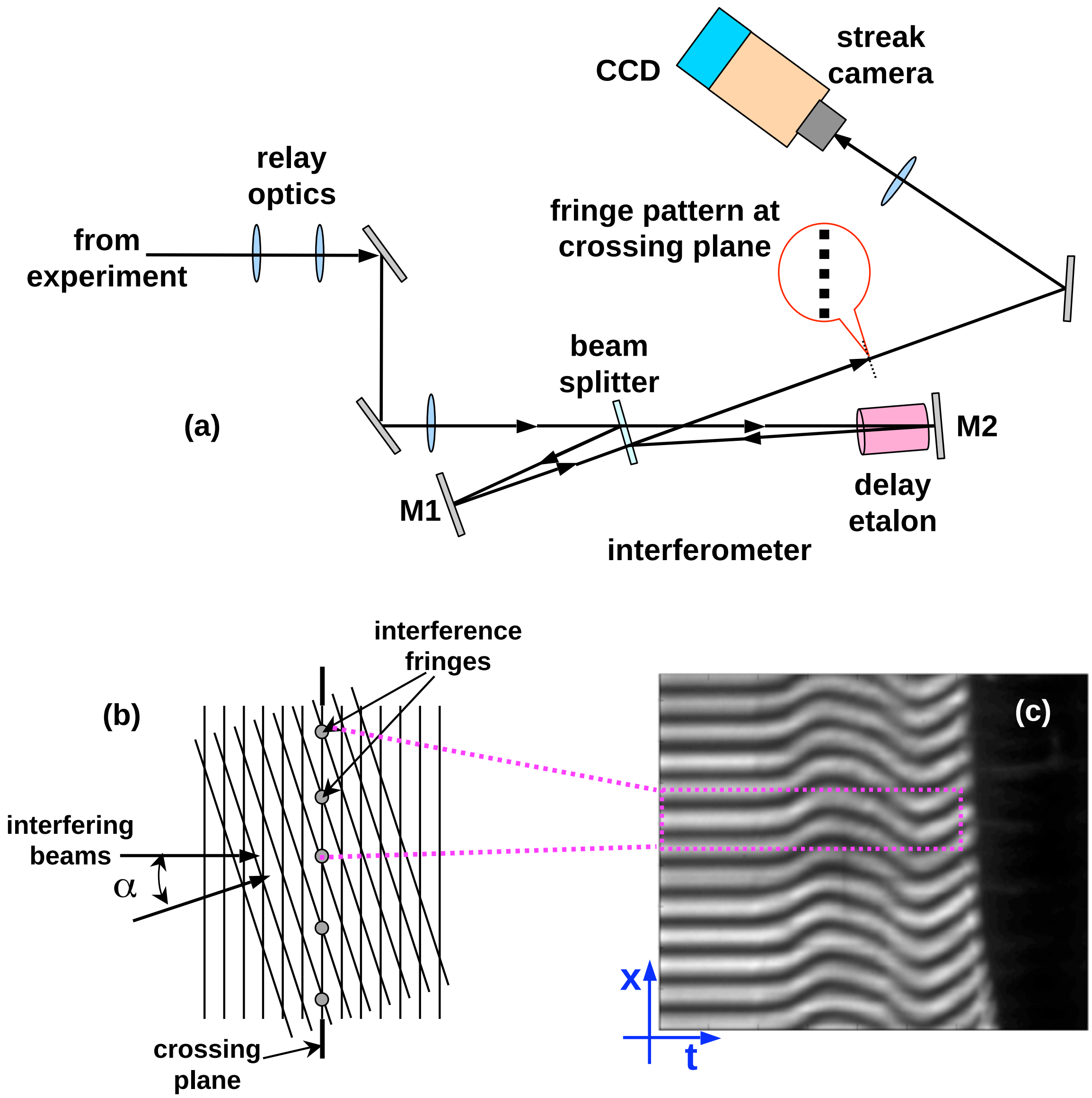

Figure 2.1. (a) Line-imaging ORVIS schematic, (b) crossing plane, and (c) streaked fringe data.

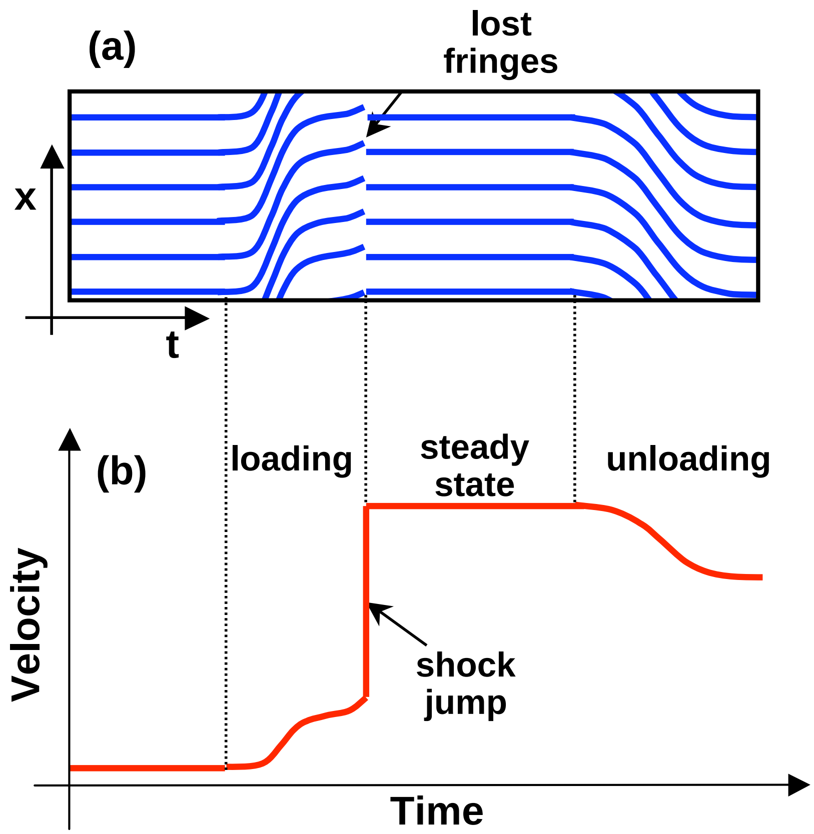

Figure 2.2. (a) Idealized fringe record, and (b) corresponding velocity history.

2.2 Fourier transform method

The Fourier transform method (FTM) was introduced by Takeda *et al.*[11], and others have developed many variations [14, 15, 16] similar to the original technique. Since there is extensive literature on the method, the following section only gives a brief description of the analysis. An overview of the Fourier transform method is presented in Fig. 2.3.

To begin, it is assumed that the recorded interferogram can be represented by a real-valued function, given by

$$S(x,t) = B(x,t) + A(x,t)\cos[\phi(x,t) + 2\pi f_0 x + \delta_0]. \tag{2.2}$$

The function $B(x,t)$ represents the slowly varying background intensity, while the function $A(x,t)$ is the amplitude of the fringes. The fringe pattern is described by a cosine whose phase consists of two parts. The $2\pi x f_0 + \delta_0$ term represents the carrier wave with a spatial frequency f_0 and an arbitrary phase offset δ_0. For the purpose of this discussion, fringe pattern is assumed to maintain a constant fringe frequency f_0. However, in reality f_0 may vary slightly over x due to imperfections in the interferometer optics and distortions in the streak camera. The term $\phi(x,t)$ encapsulates the data, which represents a phase modulation superimposed on the carrier wave. A more convenient and equivalent representation of Eq. 2.2 is

$$S(x,t) = B(x,t) + C(x,t)\exp[i(2\pi f_0 x + \delta_0)] + c.c., \tag{2.3}$$

where

$$C(x,t) = A(x,t)\exp[i\phi(x,t)]/2. \tag{2.4}$$

The FTM begins by taking a one-dimensional spatial fast Fourier transform (FFT) of Eq. 2.3 at a fixed time to give the power spectrum,

$$s(f,t) = \mathbf{FFT}\{S(x,t)\} = b(f,t) + c(f - f_0, t) + c^*(f + f_0, t). \tag{2.5}$$

The background function $b(f,t)$ can be separated from the phase information contained in c by using a bandpass filter. The filter selects a range of the power spectrum centered around the c-lobe and zeroes out the information at all other frequencies, such as the low and high frequency noise, and the negative frequency c^*-lobe,

$$s_{BP}(f,t) = \mathbf{BP}\{s(f,t)\} = c(f - f_0, t). \tag{2.6}$$

Taking the inverse fast Fourier transform (IFFT) of the bandpass filtered power spectrum produces the complex-valued function,

$$D(x,t) = \mathbf{IFFT}\{s_{BP}\} = C(x,t)\exp[i(2\pi f_0 x + \delta_0)]. \tag{2.7}$$

From this complex-valued function, a "wrapped" phase can be extracted,

$$W[\phi(x,t) + 2\pi f_0 x + \delta_0] = \arctan\left[\frac{\mathbf{Im}\{D\}}{\mathbf{Re}\{D\}}\right], \tag{2.8}$$

which is bounded in the interval $[-\pi, \pi]$. When the phase passes through odd multiples of π discontinuous jumps of π occur, which are removed using a simple unwrapping routine. Also, the underlying background phase $(2\pi x f_0 + \delta_0)$ is subtracted to give a true extracted phase. Finally, applying the velocity-per-fringe (VPF) of the line-imaging ORVIS to the extracted phase yields the velocity profile.

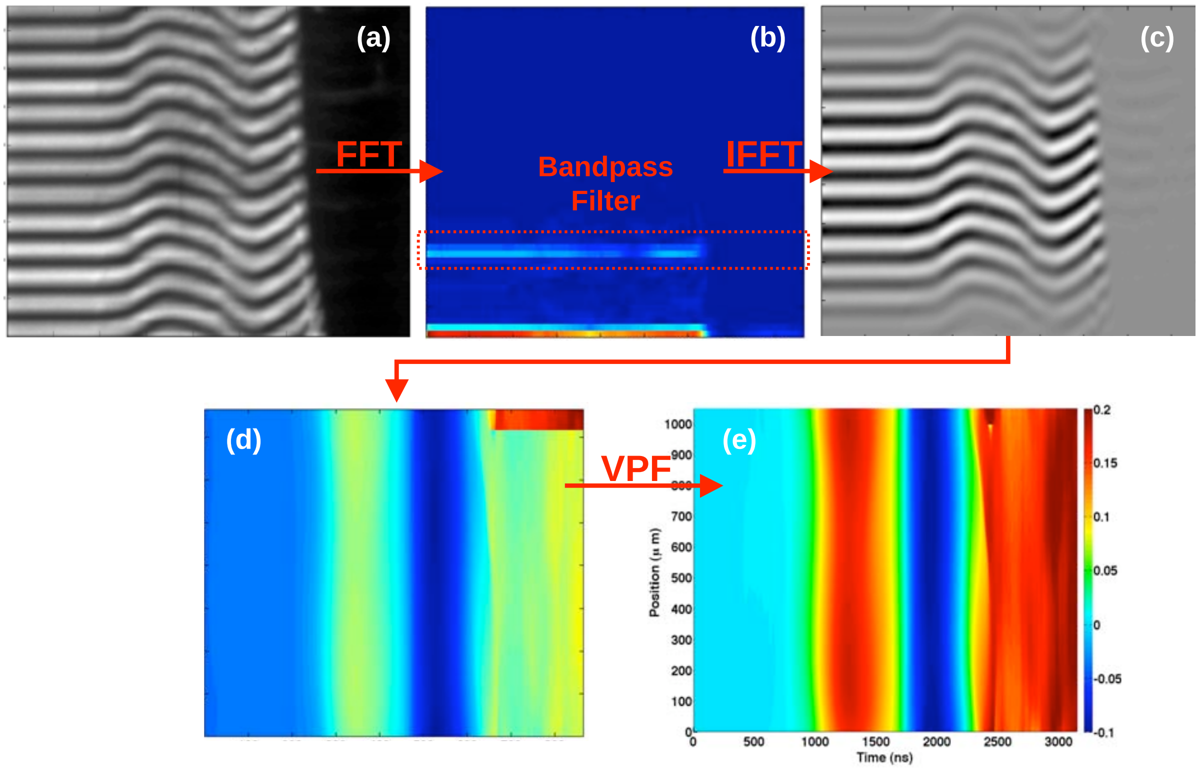

Figure 2.3. Fourier transform method: (a) raw data, (b) power spectrum with bandpass filter, (c) filtered data, (d) phase map, and (e) velocity surface.

Chapter 3

Program overview

An overview of the LIVA program is presented in this chapter. First, program installation and execution instructions are given. Next, the features of the graphical user interface are presented. Finally, the program's analysis stages are defined.

3.1 Installing and running LIVA

LIVA exists in two formats:

1. a MATLAB version which runs within the MATLAB program, and
2. a Windows executable version which may be used on Windows systems without the MATLAB program.

Installation of each version is different and are described separately in the following sections.

3.1.1 MATLAB version

The MATLAB version of LIVA is intended for MATLAB version 2008a or later. A valid MATLAB license (http://www.mathworks.com) is required. First, copy the folder `matlabLIVA`, which contains all files and subdirectories of the MATLAB version of LIVA to a local directory of the machine. Next, startup the MATLAB program. To install LIVA, add the `matlabLIVA` directory to the MATLAB path, using either the"addpath" command or the "Set path" tool on the "File" menu. Only the main folder itself, not the `private` subdirectory, should be added to the path. After installation is complete, the program can be started by typing "LIVA" at the command line.

3.1.2 Windows executable

The executable version of LIVA is intended for Windows XP. The program may operate in older versions of Windows but is not supported. Also, the executable version of LIVA has not been tested on Windows Vista but should presumably work. After the contents of the `winexe` folder has been copied to a local directory of the machine, double click the `MCRInstaller2008a.exe` program and accept all default choices in the installation. This process installs necessary libraries and support functions for LIVA, and needs to be performed once for each machine where the program is to be used. When the MCR installer is complete, LIVA can be launched by double clicking on the `LIVA.exe` executable. The initial launch of the program will be somewhat slow as the various routines are unpacked for the first time, but subsequent launches should be considerably faster.

3.2 Graphical user interface

This section describes the graphical user interface (GUI) of LIVA, as shown in Fig. 3.1. The LIVA GIU consists of a main figure, buttons, menu, and tools.

3.2.1 Main figure

In the center of the LIVA GIU is the “main figure”, where the majority of the user graphical interactions occur. As the user proceeds through the analysis stages, the main figure will display the appropriate image or plot.

3.2.2 Buttons

The “buttons” are grouped into separate panels corresponding to the analysis stages described in the analysis overview, section 3.3. Clicking on a button will initiate the labeled analysis function. Dialog boxes will appear to guide the user through the chosen analysis routine until it’s completion.

3.2.3 Menu items

At the top of the LIVA GUI is the “menu” bar, which consists various actions that are grouped into the headings: “Load Data”, “Save Data”, “Figures”, and “Program”. Upon clicking on a menu heading, a list of action is revealed.

Under the Load Data menu heading, the following actions are enabled: “Set Directory”, “Import Data”, and “Import Configuration”. The first action sets the directory from which input data files will be loaded into LIVA, and where output files will be saved. Upon clicking the import data action, the user is asked to load 4 image (.tiff or .tif) files to be individually into the program. The import configuration action allows the user to recover saved parameters of previous work done on the LIVA program.

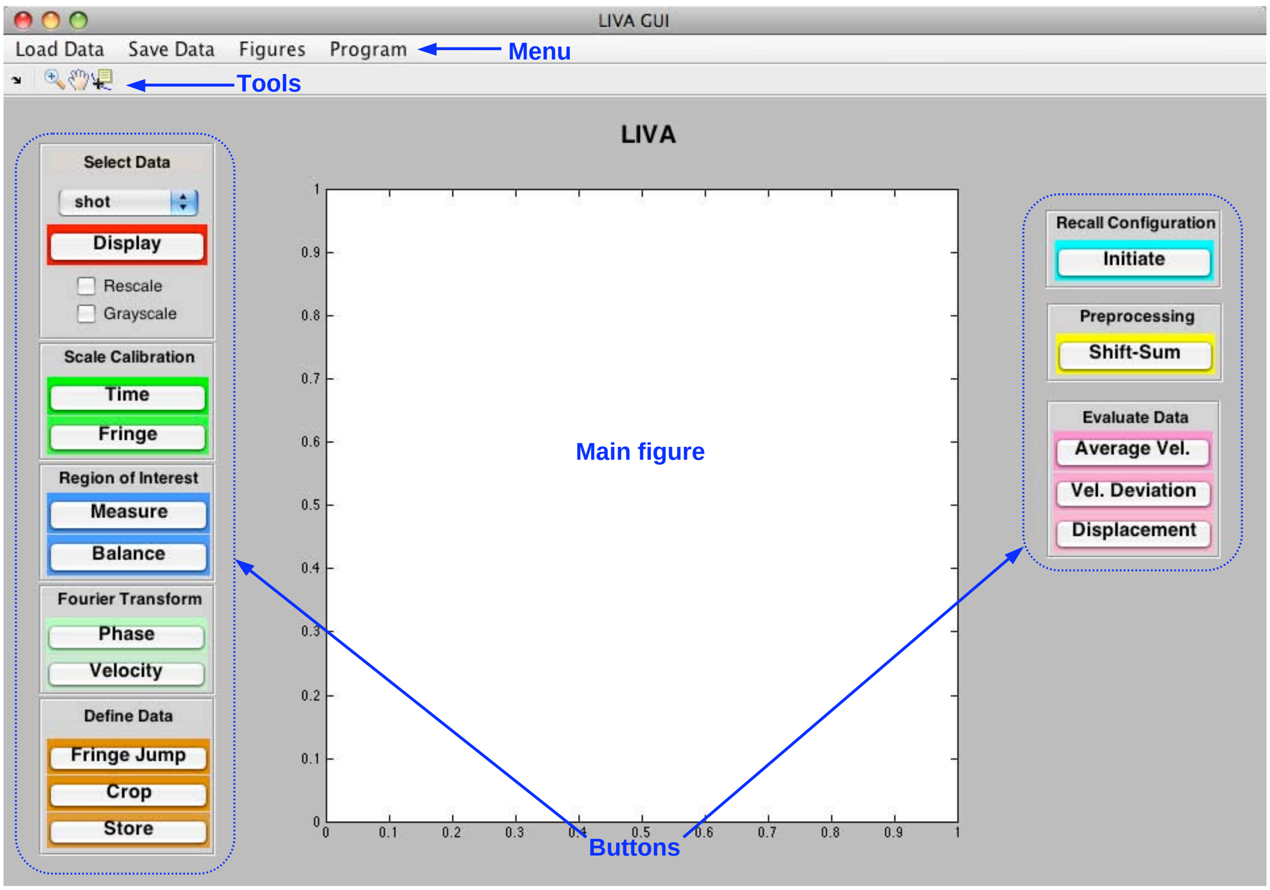

Figure 3.1. LIVA GUI.

Under the Save Data menu heading, the following actions are enabled: “Export Data”, and “Export Configuration”. The export data action saves the fully analyzed velocity matrix (as function of time and space), average velocity, velocity deviation, and displacement into ASCII text files. The export configuration action allows the user to save parameters of the current work such as input data filenames, scale calibrations, region of interest, bandpass cutoffs, velocity-per-fringe, etc. into an ASCII text file.

Under the Figures menu heading, individual images and plots may be opened in separate figure windows. The figures can be save to a file using the “Save figure” button (disk icon) of the figure window. The default format is a MATLAB figure (.fig), but other formats (.pdf, .eps, .jpg, .tif, etc.) are available.

Under the Program menu heading, the following actions are enabled: “About”, “Restart”, and “Exit”. The about action provides general information about LIVA. Restart closes and relaunches the program, clearing all parameters and returning the program to its default state. Exit closes the program.

3.2.4 Tools items

Under the Menu is the “Tools” bar, which contain the following tools to aid in the analysis.

1. “Zoom” - plots can be zoomed into with either a left mouse click or by clicking and dragging to create a zoom box. Shift-click zooms out and double click restores to the original view. Press the right mouse button for additional options.

2. “Pan” - user can click and drag the plot around for better viewing.

3. “Data cursor” - displays the (x, y, z) values of the point that the user clicks on. Right-click for additional options.

3.3 Analysis overview

The analysis stages of the LIVA program are displayed in Fig. 3.2, which are grouped as either “primary” or “optional” stages. The primary stages represent the core functions of the program, while the optional stages are supplementary functions that may be performed in conjunction with the primary stages.

First, the data is loaded into the LIVA program, and then selected for data reduction. Initial scale calibrations of time and fringe spacing are performed. Next, the region of interest of the data is cropped. Using the Fourier transform method, the phase is extracted and then converted to velocity. The final velocity data is defined and saved.

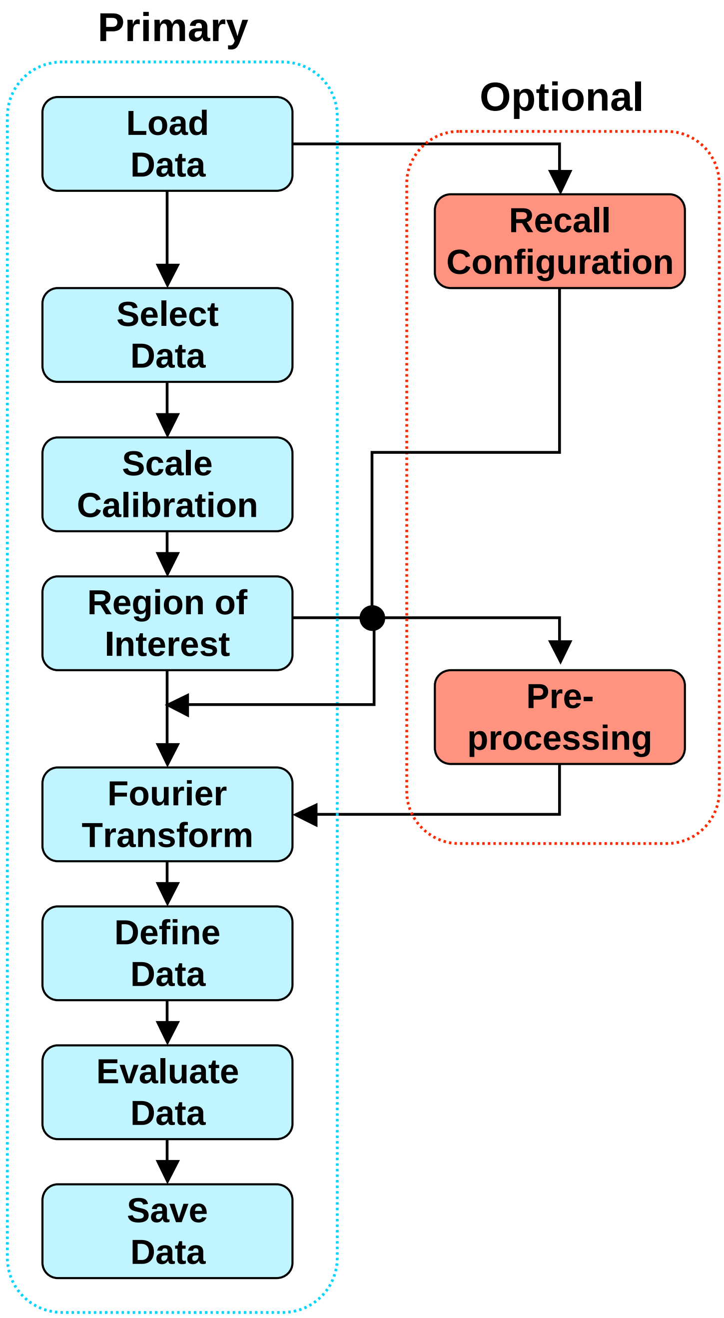

Figure 3.2. LIVA operational layout.

3.3.1 Primary stages

Load data

The LIVA program begins by opening the load data menu. The first step is to set the directory for importing and exporting files. For a new round of analysis, the import data action is used to load new data image files. LIVA expects the following 4 data image files:

1. "shot" - image of fringes during the experiment,
2. "preshot" - image of fringes prior to the experiment,
3. "background" - image with the streak camera shuttered, and
4. a blocked" - image with one leg of the VISAR interferometer blocked (either reference or delay).

The orientation of the images should be such that the data fringes run parallel to the horizontal axis, and with the time increasing from left to right. After loading, the user is asked if the images requires any rotation (90°) or flipping (horizontal and/or vertical) in order to comply with the required orientation.

To resume a previous round of analysis, click on the import configuration action to load a configuration file (.config) which contains information on previous work parameters and data files. Refer to the recall configuration section on how to continue previous work.

Select data

Within the select data panel are the "Images" pop-up list, "Display" button, and "Rescale" and "Grayscale" check boxes. To view a data image, click on the pop-up list and choose one of the listed images: "shot", "preshot", "background", "blocked", or "ROI" (region of interest), and then click on the display button. The image will appear in the main figure with a default scaling of 0 to the image maximum value, and the "jet" colormap.

To manually change the scaling, right-click the colorbar on the right side of the main figure, and choose "Launch colormap editor". Alternatively, the image colormap is automatically rescaled for better viewing by checking the rescale check box. Also, checking the "grayscale" box will convert the colormap to a grayscale. For the check boxes to take effect, the display button needs to be clicked again.

Scale calibration

The scale calibration panel consists of the "Time" and "Fringe" calibration buttons. Normally, the shot image will contain time-comb impulses which are separated by a known time duration. After clicking on the time calibration button, the user is asked to select a box around the time-combs. The program then calculates the time spacing (pixels/cycle), which is shown in a side figure window. Next, the user is asked to input the time duration between time-combs (ns).

Similarly, upon clicking the fringe calibration button, the user is asked to select a box around the region of the image (left side) where there is no movement of the fringes. The program finds the fringe spacing (pixels/cycle), and a side figure window is shown. Finally, the program asks the user to input the fringe separation (μm), which should be measured during the experiment.

Region of interest

The region of interest panel consists of the "Measure" and "Balance" buttons. After clicking the measure button, the user is asked to select a region of interest (ROI). The ROI is set to be a square box, and it should not include the time-combs. Additionally, areas where no useful information may be obtained, such as at the late times of the experiment, should be cropped so that the analysis may run faster.

Upon clicking on the balance button, 3 actions are done to normalize the shot image prior to the subsequent Fourier transform analysis. First, the shot image is subtracted by the background image. Second, time-independent intensity variations due to the laser pulse are normalized by dividing the shot image by the blocked image. For some shot images, even after these 2 actions are done, time- and space-dependent intensity variations may remain. The user is asked to select a low pass filter to create a low frequency "form-factor" of the shot image. Finally, the shot image is divided by the form-factor image to further normalize the shot image.

Fourier transform

The Fourier transform panel consists of the"Phase" and "Velocity" buttons. Upon clicking the phase button, the user is asked to define the lower and upper frequency cutoffs. The phase shift of the shot image is extracted and shown in the main figure and in a side figure. To convert the phase shift to the velocity, the user clicks on the velocity button, where the user is asked to input the velocity-per-fringe (VPF) constant. The velocity profiles at every vertical pixel location is then shown in the main figure.

Define data

The define data panel consists of the "Fringe Jump", "Crop", and "Store" buttons. The fringe jump button allows the user to add or subtract fringes to the velocity profiles. If there are no fringe jumps, this button can be skipped. Upon clicking the fringe jump button, the user is asked to define the horizontal window around where the fringe jump needs to occur. Also, the user can define the vertical window around where the fringe jump occurs. If one or more fringes need to be added then a positive fringe number is inputted. Conversely, if one or more fringes need to be subtracted then a negative fringe number is inputted.

Upon clicking the crop button, the user is asked to set the horizontal start and end of the velocity profiles. At this time, the program calculates the velocity profiles of the preshot image as well. The user is asked if the "preshot velocity" needs to be subtracted from the "shot velocity". The program uses the earlier time calibration to convert the horizontal pixel locations to time points.

The velocity profiles may now be stored into a data matrix which is can be used for plotting, further analysis, and saving. Upon clicking the store button, the user is asked to input a number (0-9) under which to store the velocity data matrix.

Evaluate data

In the evaluate data panel, 3 additional functions are included in the LIVA program. Upon clicking the "Average Velocity" button, the user is asked to input the number of the stored velocity data matrix for averaging. The averaging is done across the spatial direction, and a single velocity profile is displayed in the main figure. This average velocity profile is then used to find the velocity deviation upon clicking the "Velocity Deviation" button. Finally, clicking the "Displacement" button, the program calculates the how far the measured material surface moves during the experiment.

Save data

Upon clicking the export data action under the save data menu, the user is asked to choose the number (0-9) of the stored data matrix which the user wants to export. The data matrix is saved under the default file name, "LIVAvelocity.txt". The corresponding time (ns) and position (μm) vectors are saved under the default filenames, "LIVAtime.txt" and "LIVAposition.txt", respectively. Other data such as average velocity, velocity deviation, and displacement are saved under the default filenames, "LIVAaverage_velocity.txt", "LIVAvelocity_deviation.txt", and "LIVAdisplacement.txt", respectively.

3.3.2 Optional Stages

Recall configuration

After loading a configuration file, the user clicks on the "Initiate" button in the recall configuration panel. Based on the information in the .config file, the program loads the appropriate input files, creates the specified region of interest, and sets the required work parameters. The user may now continue at the Fourier transform analysis stage of the program.

Preprocessing

For the case of unusually poor quality data, additional preprocessing of the data may be warranted. Upon clicking the "Shift-Sum" button in the preprocessing panel, the user is asked to input the number of times to shift-sum the image.

Chapter 4

Using LIVA

This chapter describes the practical use of the LIVA program. Analysis of two examples are given with step-by-step results for the user to compare with. Two sets of line-imaging ORVIS data have been provided with the LIVA package, and are located in the folder `LIVAexamples`. The first example, (`LIVAex1`) represents data of reasonable image quality which requires a minimal amount of work to obtain the velocity information. However, the second example (`LIVAex2`) represents data of low image quality, which require additional preprocessing and more careful analysis to produce meaningful results.

4.1 Example 1

Example 1 is taken from an experiment performed on the Veloce pulsed power generator [7]. Line-imaging ORVIS was used to measure the velocity of a 20 mm wide by 1.5 mm thick Cu panel, backed by a 8 mm thick Sapphire window, as it was ramp loaded up to 70 kbar using a 440 ns, 2.5 MA current pulse. The data images are contained in the 4 TIFF image files: `LIVAex1_shot.tif`, `LIVAex1_preshot.tif`, `LIVAex1_background.tif`, and `LIVAex1_blocked.tif`. After setting the directory to where the image files are located, load each data image file with the load data action. For this set of data, at the "Image Manipulation" dialog answer: "y", "y", and "n". Figure 4.1 shows the 4 data images after being loaded into the LIVA GUI.

After clicking the time calibration button, make a box around the time combs located at the top of the shot image (Fig. 4.2), and then input 200 ns for the comb separation, see Fig. 4.3a. Similarly, after clicking the fringe calibration button, make a box around straight fringes on the left side of the shot image (Fig. 4.2), and then input 90 μm for the fringe separation, see Fig. 4.3b.

Next, click on the measure button and make a box around the region of interest for analysis. Figure 4.4 displays the region of interest cropped with vertical pixel locations from $pix_{vi} = 50$ to $pix_{vf} = 850$, horizontal pixel locations from $pix_{hi} = 100$ to $pix_{hf} = 900$. A side figure of the ROI (not shown) is also given for the user to examine separately. After clicking on the balance button, normalization of the data is performed. The default value for the horizontal and vertical form factors is 0, which correspond to having only background image subtraction and blocked image division done on the shot image. Entering values greater than 0 creates a form factor image for division of the shot image. Figure 4.5 displays the balancing of the ROI with a horizontal factor HFF = 50, and a vertical form factor VFF = 40.

Click on the phase button to begin phase extraction using the Fourier transform method. The user chooses either “old” or “new” bandpass cutoff parameters. The first choice uses values obtained in previous attempts at phase extraction, while the second choice allows the user to select points from the power spectrum of the fringes, as shown in Fig. 4.6. The fringe frequency is centered at about $f_{center} = 13$, and by inputting frequency cutoffs at $f_{low} = 9$, and $f_{high} = 17$, the low and high frequency noise components of the image are eliminated. To avoid edge effects of the FT analysis, both the top and bottom portions of the image are cropped by 25 pixels, and results in the extracted phase shown in Fig. 4.7. Next, after clicking on the velocity button and entering a VPF = 0.1938 km/s/f, the program produces velocity profiles from the shot image, as shown in Fig. 4.8.

Sometimes discontinuities might occur in the velocity profiles due to jumps in the fringes. In such cases, the fringe jump button may be used to add or subtract fringe (velocity) jumps, and is presented in second example. Otherwise, the next step is to choose the desired time range for further analysis. Since there is no velocity information at early times and at late times it is just noise (Fig. 4.8), click on the crop button and choose the initial time cutoff $t_i = 50$ and the final time cutoff $t_f = 550$. At this time, the velocity of the preshot image is also calculated. If the fringes drift vertically then the preshot velocity is used to correct this ”drift” velocity in the shot image, but for this example pick “No”. Figure 4.9 presents the cropped velocity profiles. Now, the velocity profiles may be stored within the program by clicking on the store button and inputting the store data number “0”.

Click on the average velocity button to find the spatially averaged velocity of the velocity profiles, as shown in Fig. 4.10. The velocity deviation for this average is obtained by clicking on the velocity deviation button, and the result is shown in in Fig. 4.11. Finally, the displacement of the measured interface is calculated after clicking the displacement button, and is shown in in Fig. 4.12.

To view a list of available individual side figures to plot, click on the figures menu. After selecting a figure to plot, the user can choose to close previous figure versions (“y”) or to leave the previous figures open (“n”). Figure 4.13 display the side figures for velocity, average velocity, velocity deviation, and displacement. To close the figures, click on the close all option under the figures menu.

Finally, click on the save data menu to view the available export options. The export data option saves the analyzed data such as velocity, average velocity, velocity deviation and displacement to ASCII text files (“.txt”). The export configuration option saves the parameters of the current analysis to an ASCII text file (“.config”).

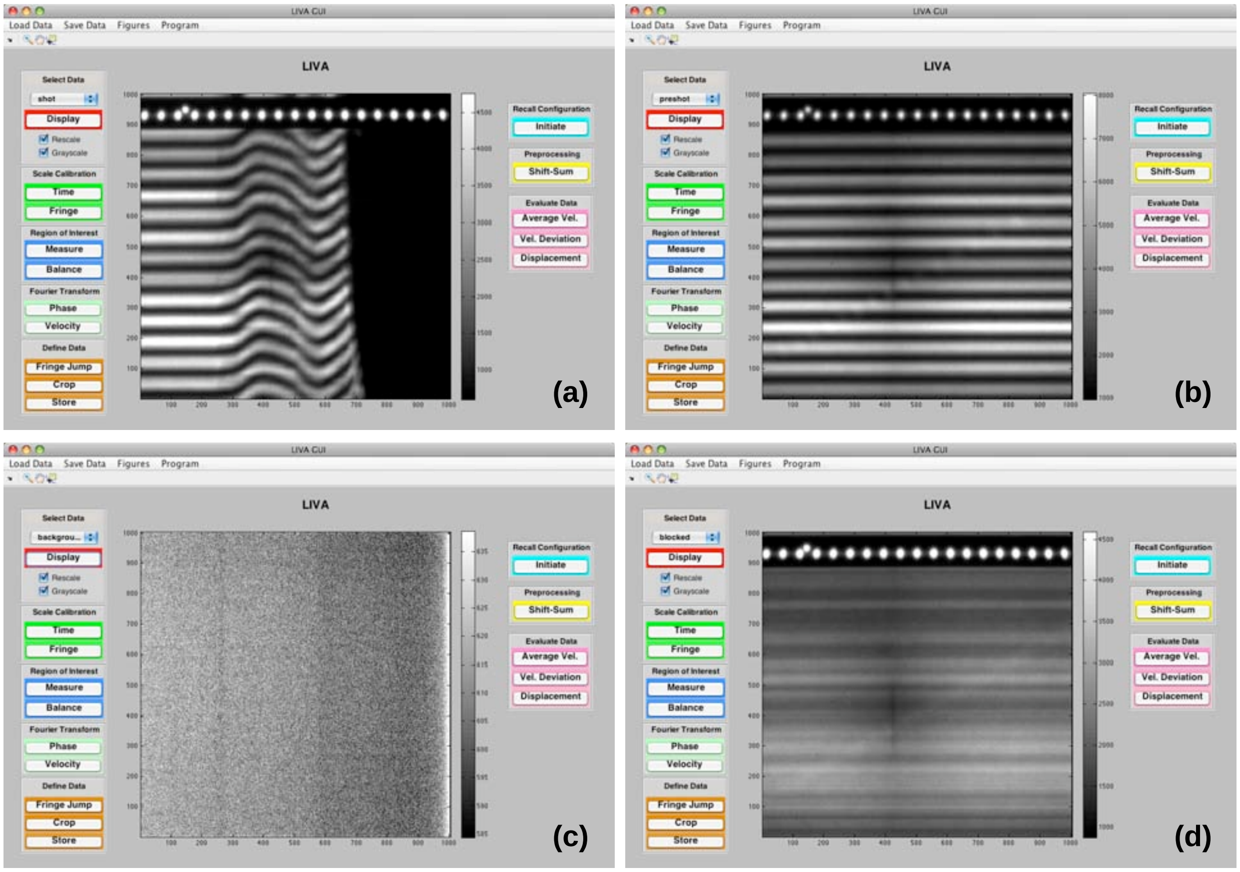

Figure 4.1. LIVA example 1: data images, (a) shot, (b) preshot, (c) background, and (d) blocked.

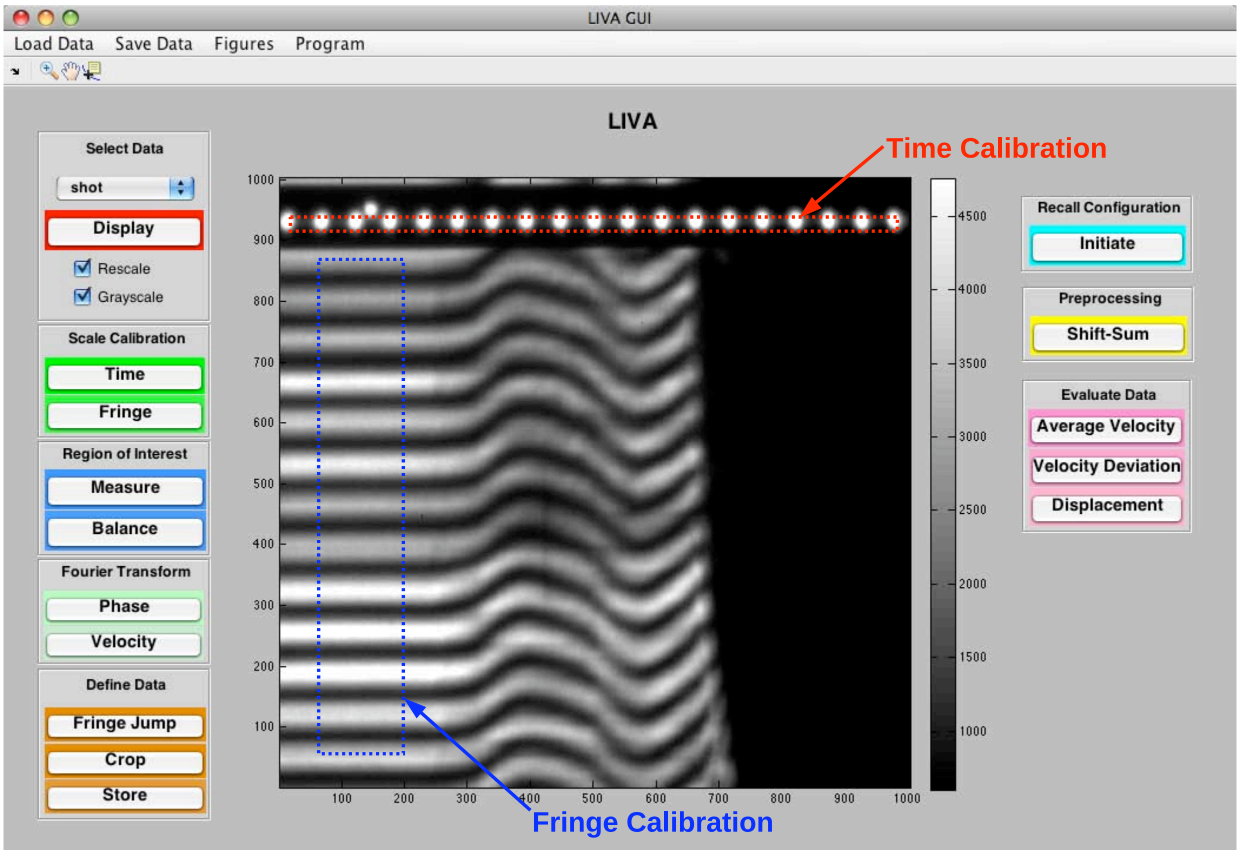

Figure 4.2. LIVA example 1: selecting time and fringe calibrations.

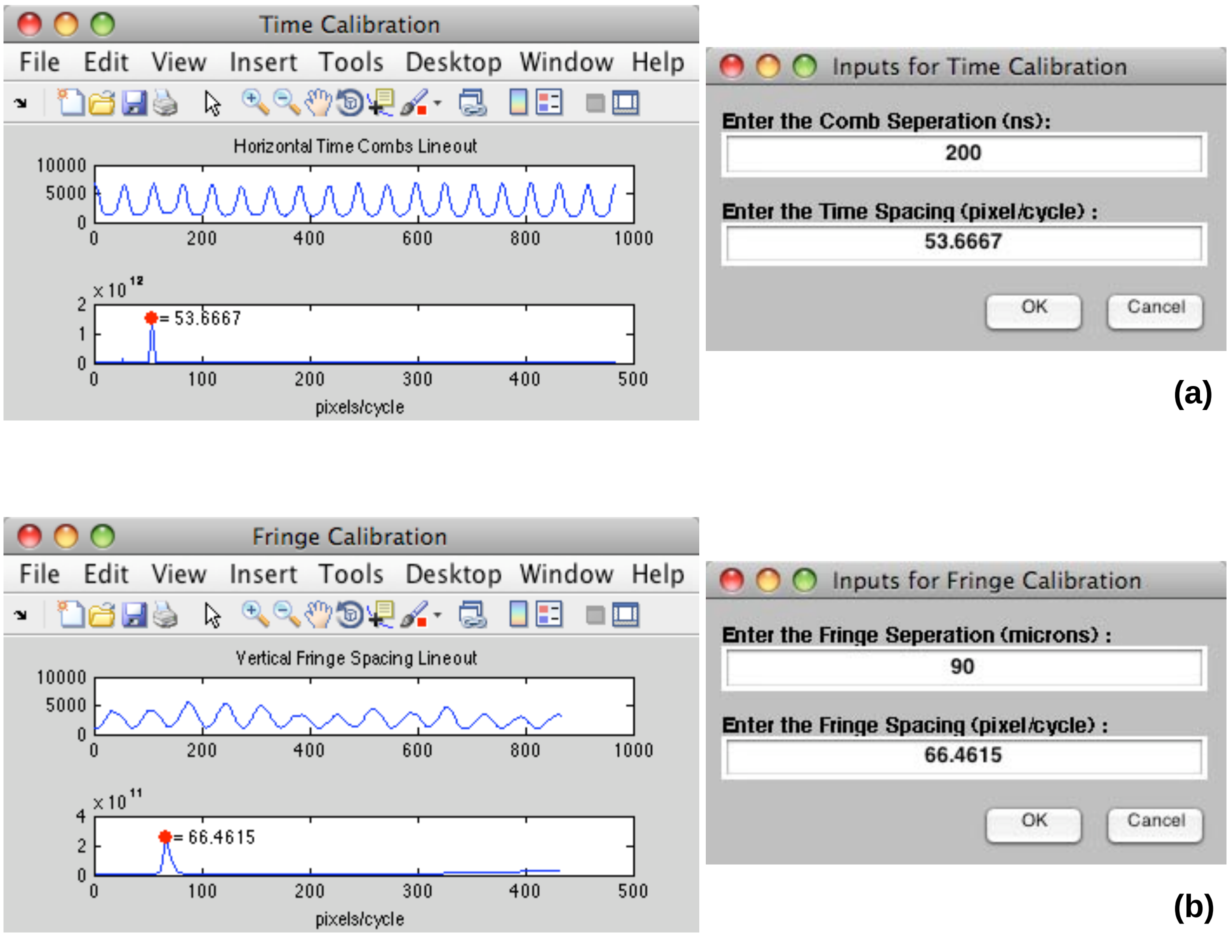

Figure 4.3. LIVA example 1: (a) time calibration, and (b) fringe calibration.

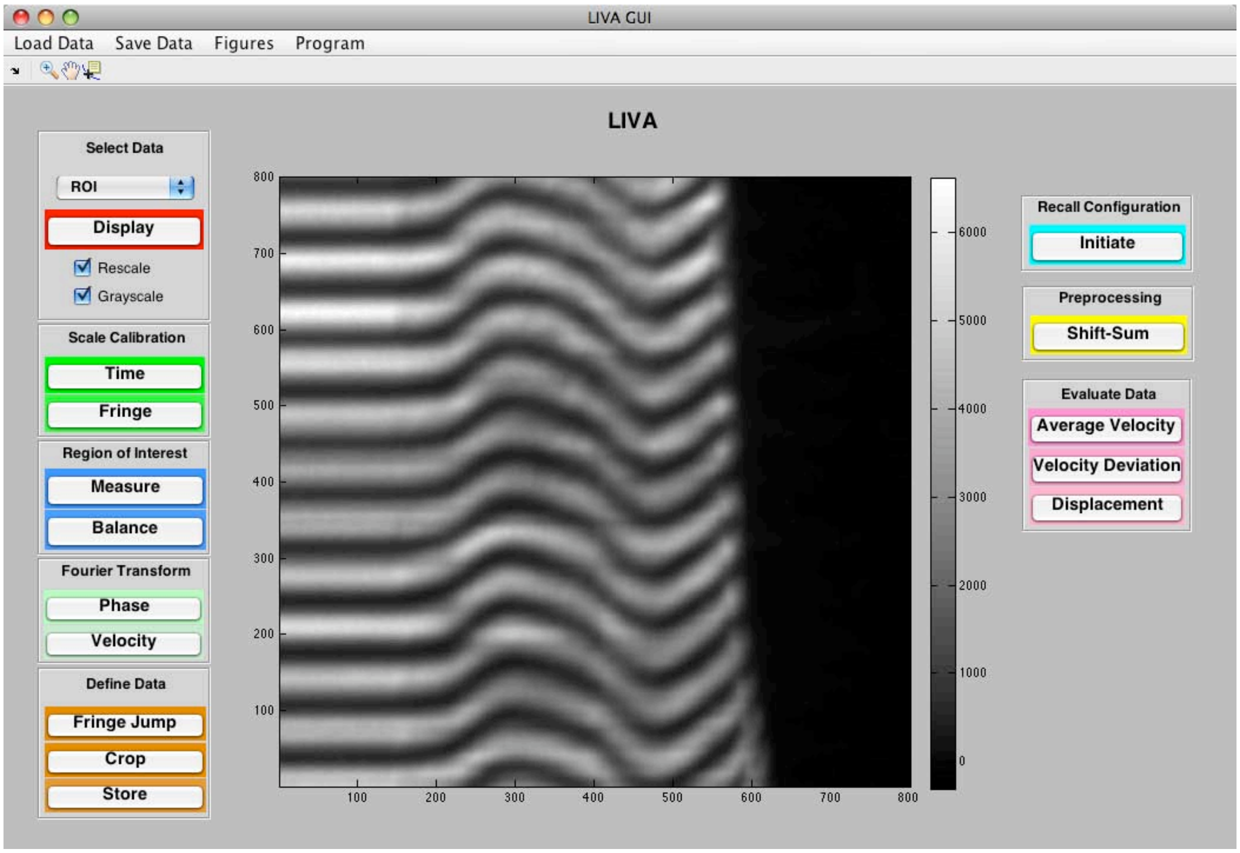

Figure 4.4. LIVA example 1: region of interest (ROI).

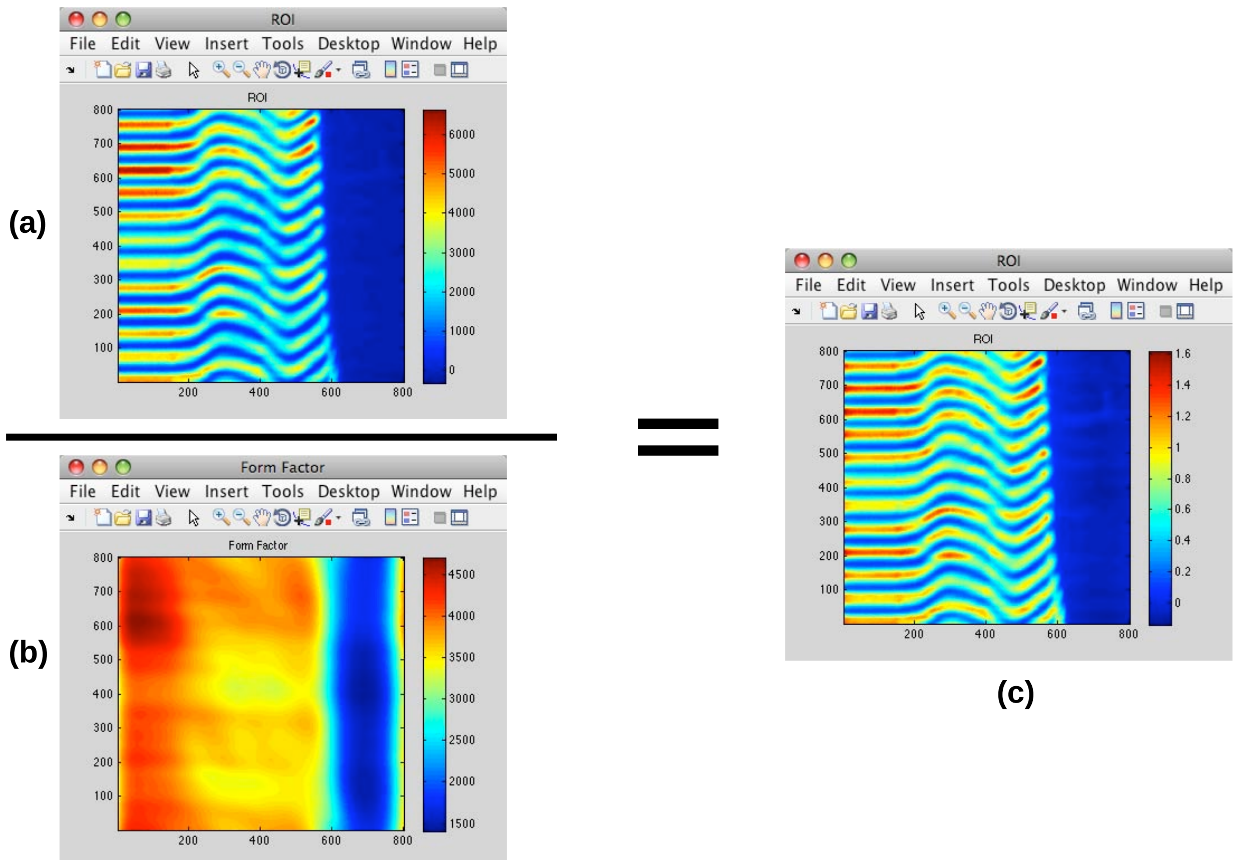

Figure 4.5. LIVA example 1: balancing data, (a) unbalanced ROI, (b) form factor (HFF=50, VFF=40), and (c) balanced ROI.

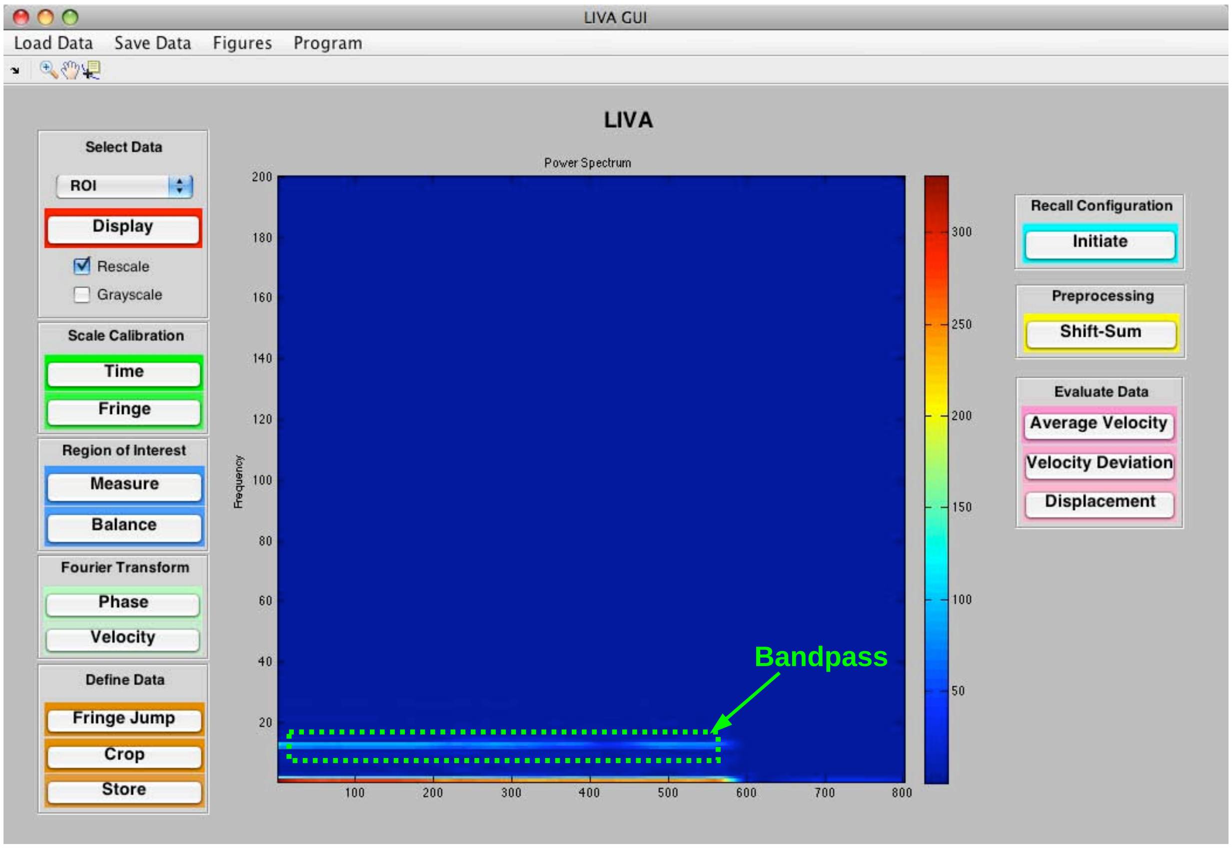

Figure 4.6. LIVA example 1: power spectrum, bandpass at $f_{low} = 9$ and $f_{high} = 17$.

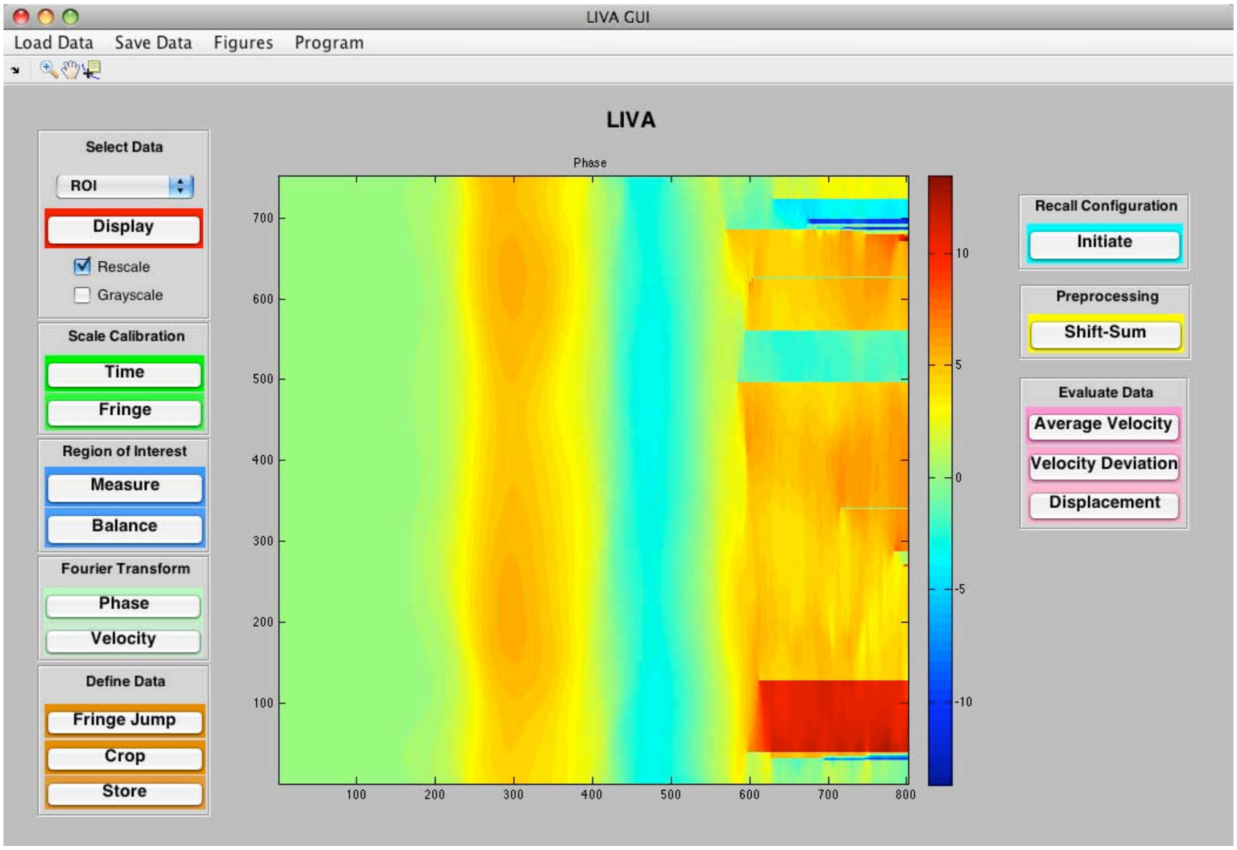

Figure 4.7. LIVA example 1: extracted phase.

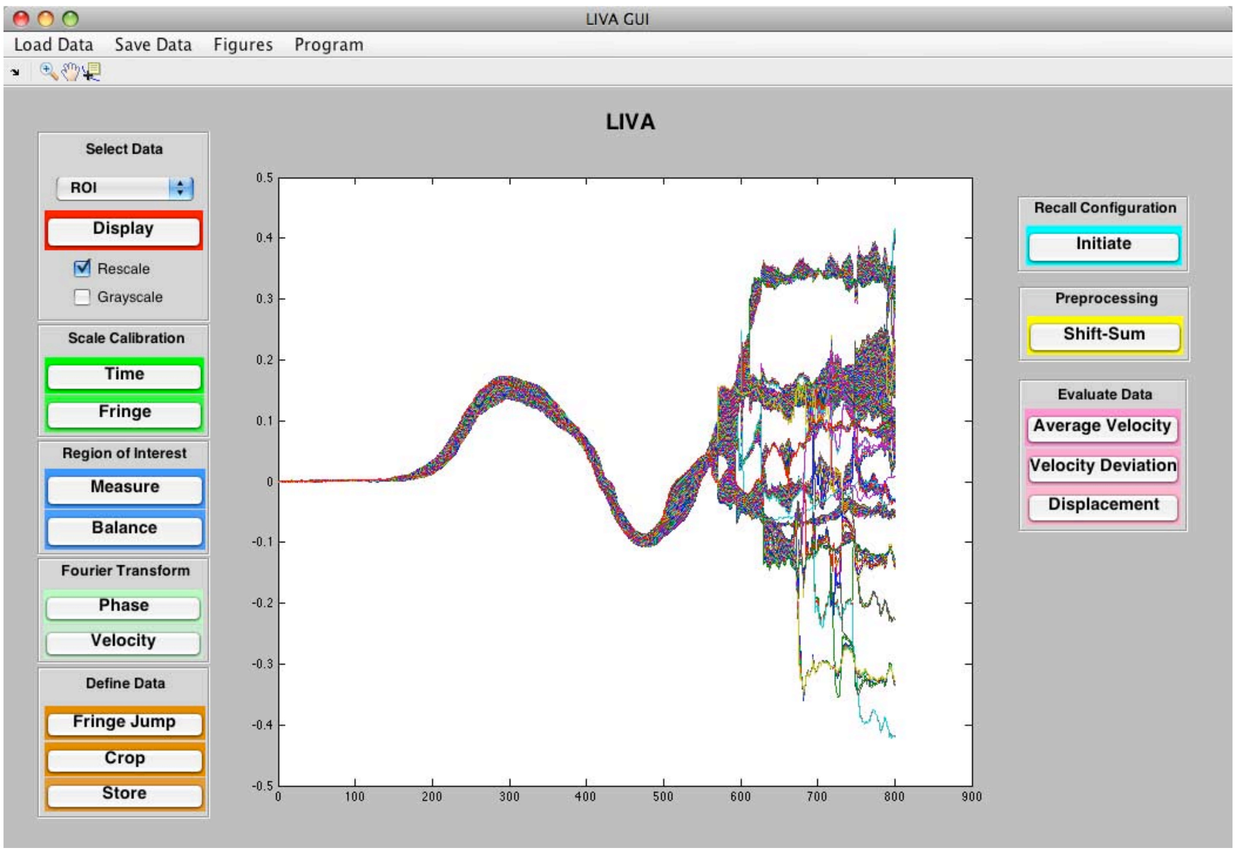

Figure 4.8. LIVA example 1: velocity profiles.

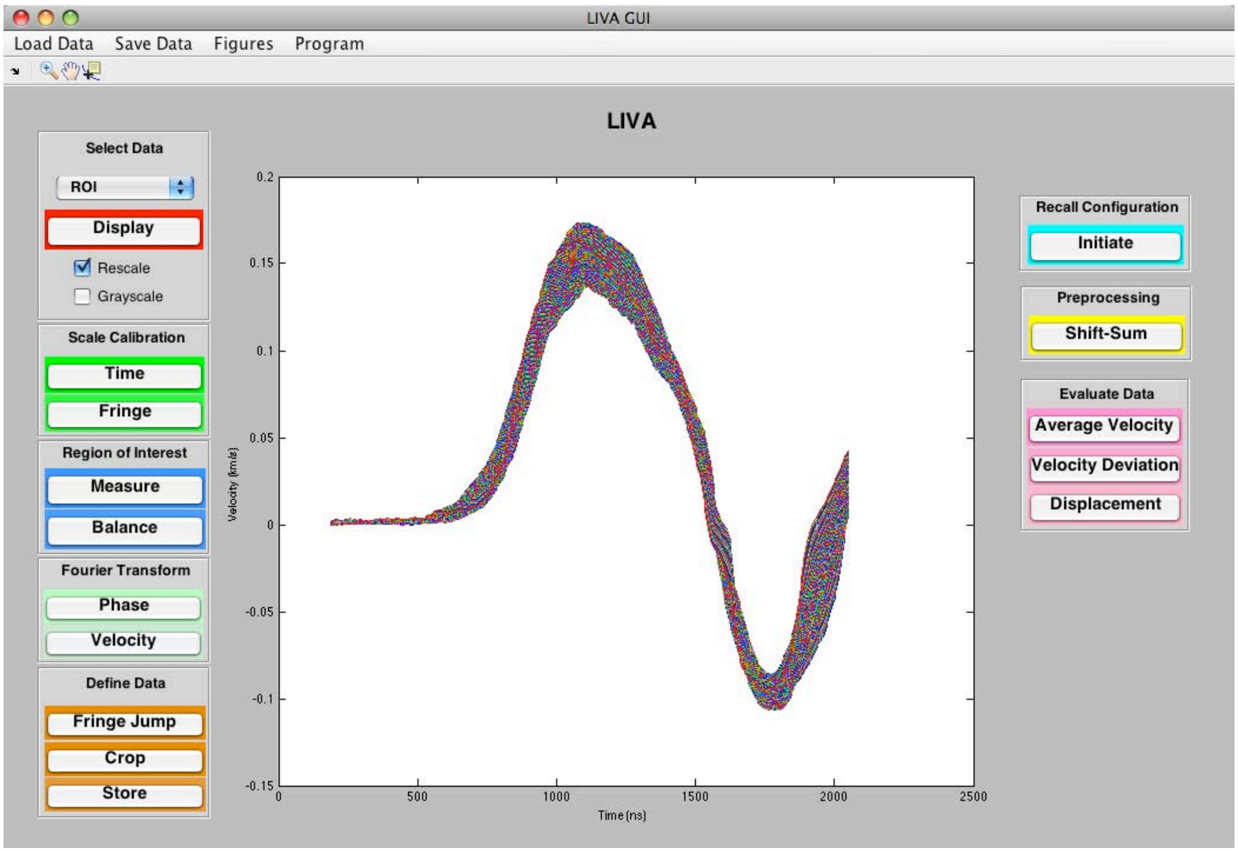

Figure 4.9. LIVA example 1: cropped velocity profiles.

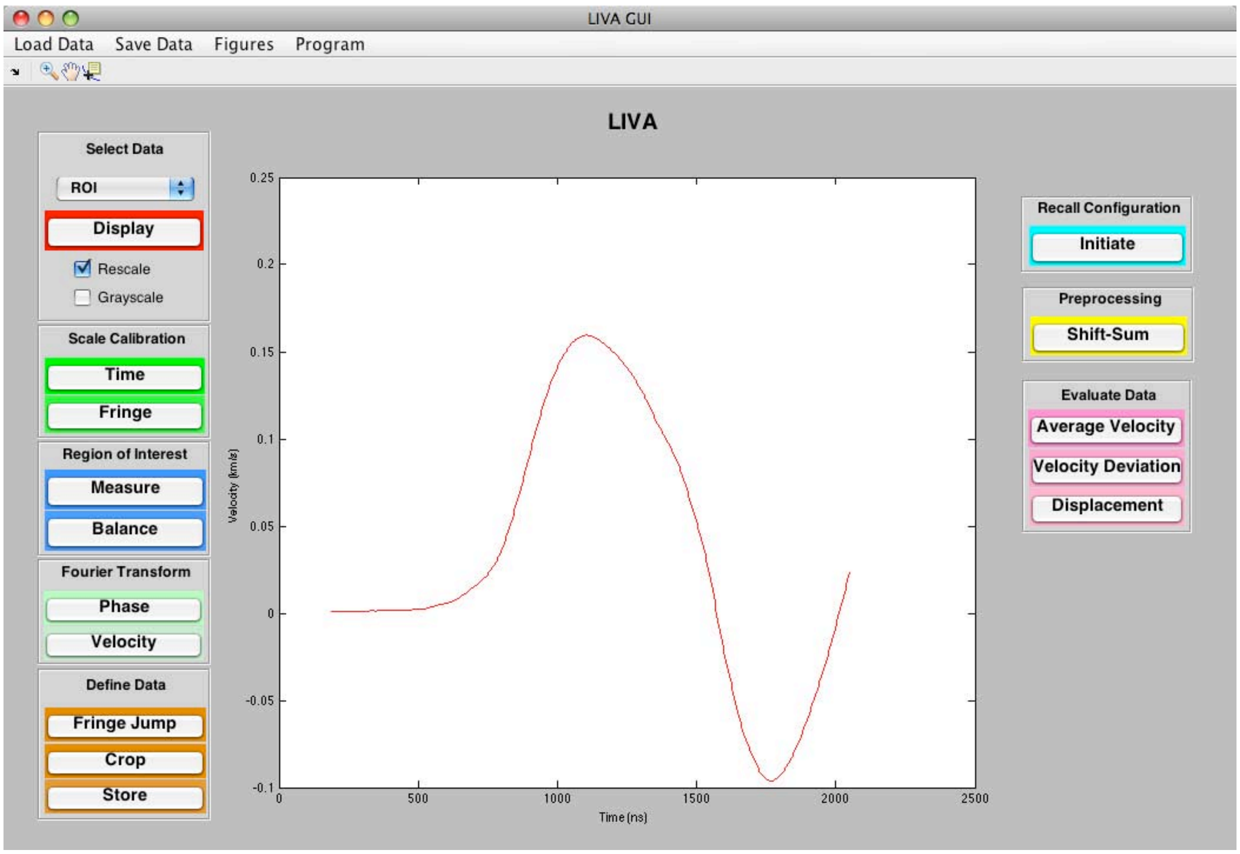

Figure 4.10. LIVA example 1: average velocity profile.

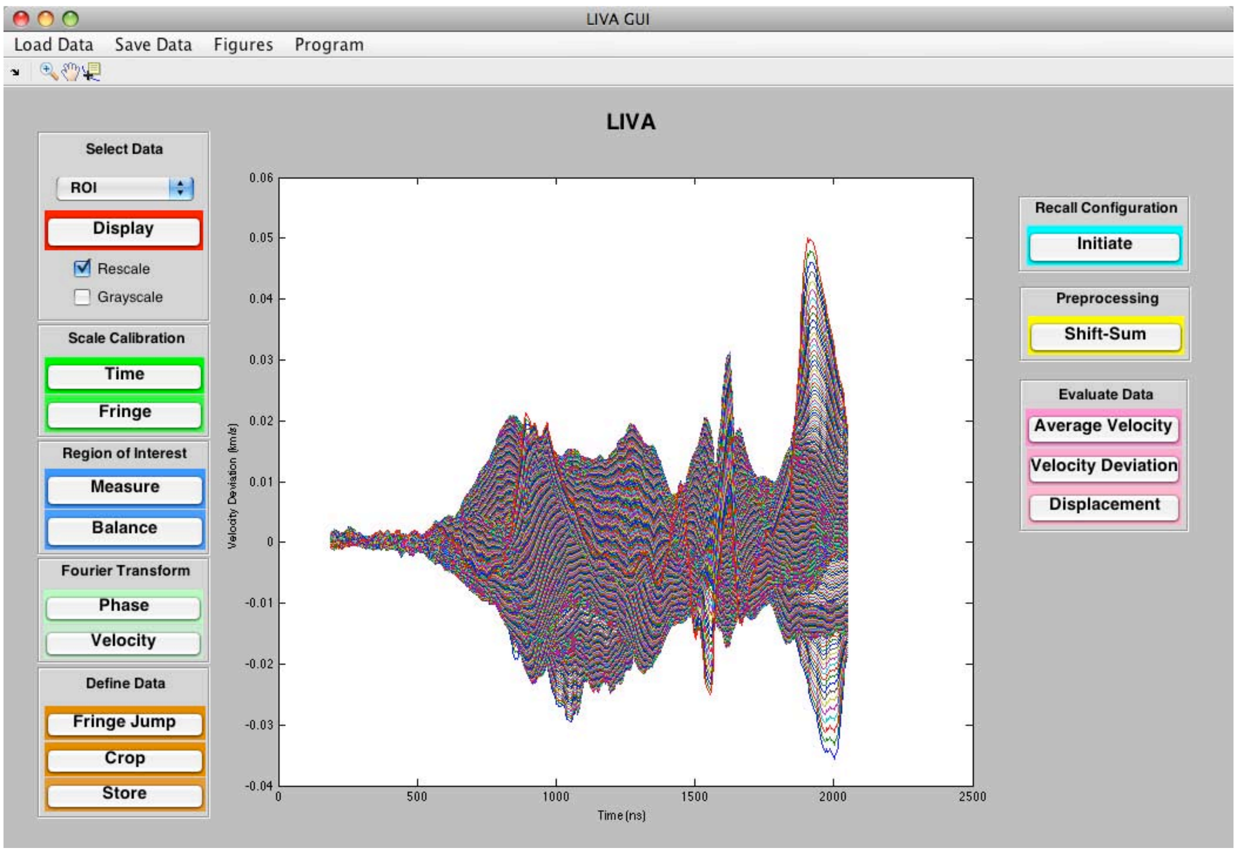

Figure 4.11. LIVA example 1: velocity deviation.

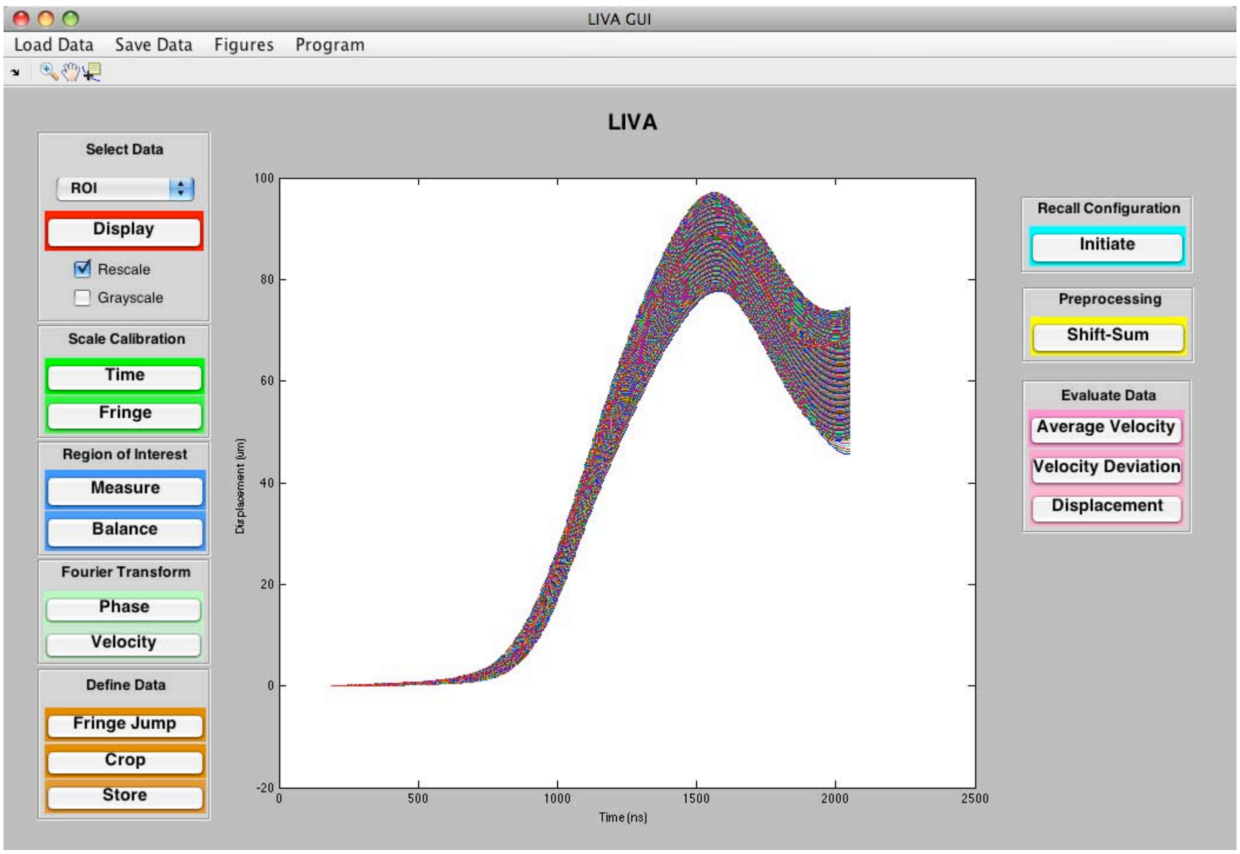

Figure 4.12. LIVA example 1: displacement.

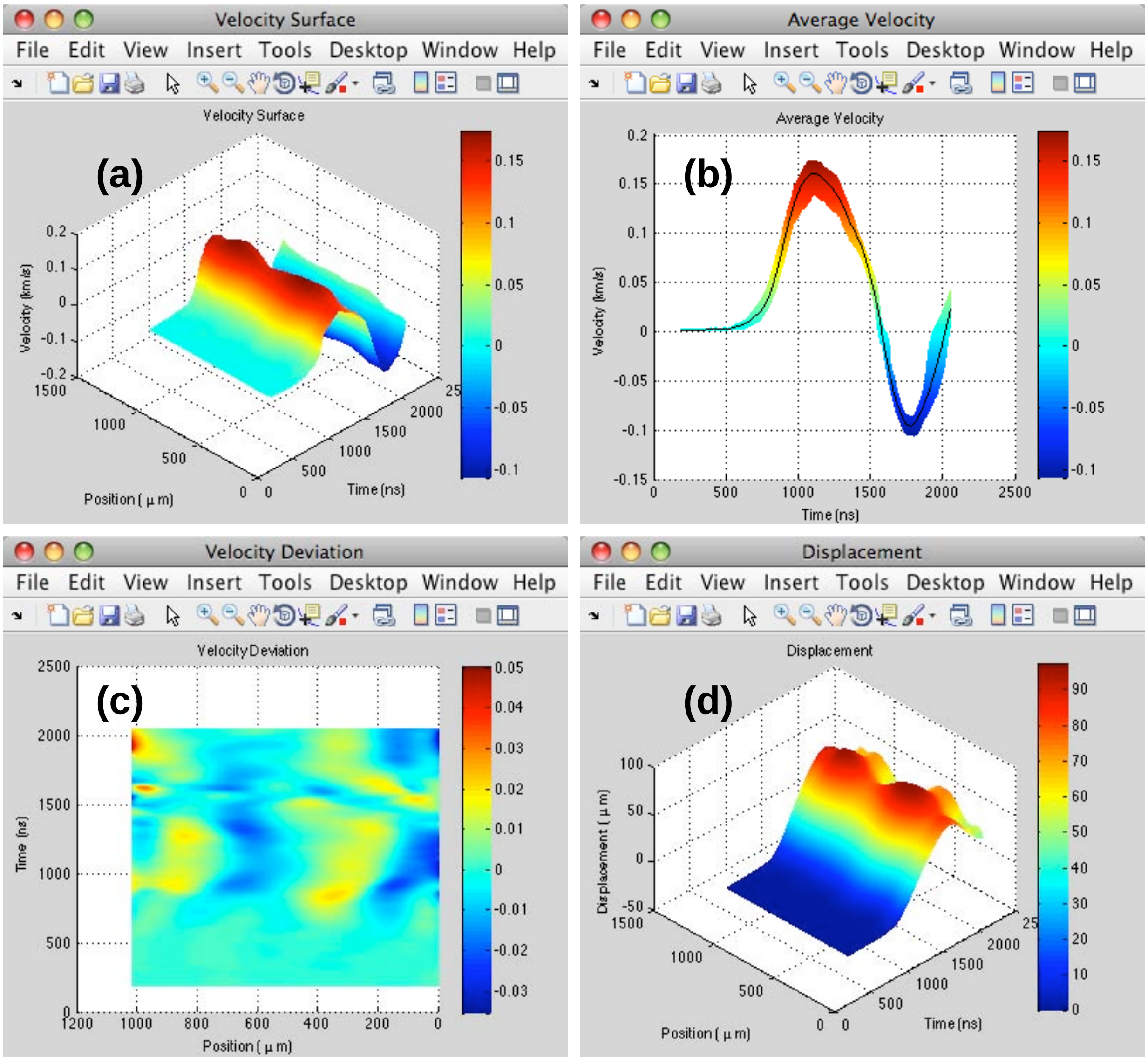

Figure 4.13. LIVA example 1: final plots of (a) velocity surface, (b) average velocity, (c) velocity deviation, and (d) displacement.

4.2 Example 2

Example 2 is taken from an experiment of a 20 mm wide by 1.5 mm thick Al panel, backed by a 8 mm thick LiF window, ramp loaded up to 70 kbar on Veloce (440 ns, 2.5 MA). The data images are contained in the 4 TIFF image files: `LIVAex2_shot.tif`, `LIVAex2_preshot.tif`, `LIVAex2_background.tif`, and `LIVAex2_blocked.tif`. Also provided for this example is the text configuration file, `LIVAex2.config`, which contains preset analysis parameters.

After setting the directory to where the image files are located, load the `LIVAex2.config` file with the load configuration action. Next, click on the initiate button to recall the parameters of the preset configuration, which also loads the data images automatically into the program. Figure 4.14 shows the 4 data images after being loaded into the LIVA GUI. The configuration file also had the preset parameters for time and fringe calibration, ROI, and balancing so the analysis may begin at the Fourier transform stage, as shown in Fig. 3.2. In this example, the shot image displays considerable "smearing" of the fringes, as shown in Fig. 4.15. After performing the phase analysis, the calculated velocity profiles are chaotic and contain numerous discontinuities, as shown in Fig. 4.16. Thus, preprocessing of the data is required to produce useable results.

The shift-sum preprocessing attempts to enhance periodic features while diminishing random noises like the smearing effect seen in this example. The procedure begins by first creating several copies of the shot image. For a correction factor of "1", one copied image is shifted up by one fringe spacing while another copied image is shifted down by one fringe spacing. Both images are then added to the original shot image and the resulting image is divided by three to produce a new shift-sum shot image. For a correction factor of "2", two more copied images, now shifted up and down by two fringe spacings, respectively, are added to the original shot image, and the total image is divided by five to obtain the shift-sum shot image. A similar procedure is performed for a correction factor of "3". However, no higher correction factors are allowed since the image would start to become "over-smoothed" and spatially dependent details of the velocity profiles would be lost.

Figure 4.16 shows the shift-sum shot image after clicking on the shift-sum button and input a correction factor of "2". Now, after redoing the phase analysis, the velocity profiles for the shift-sum data is considerably cleaner than the original data, as shown in Fig. 4.18. However, a discontinuity still occurs in some of the velocity profiles around the horizontal point $h_{fj} = 340$. Figure 4.19a displays the discontinuity more clearly. To fix the discontinuity, click on the fringe jump button and define the start and end points for the fringe jump window. Next, input a single negative fringe jump ("-1") along with the other parameters shown in Fig. 4.19b. The fringe jump corrected velocity profiles is shown in Figure 4.20. The users is left to crop, store, evaluate, and export the data for this example. Finally, click on the export configuration under the save data menu to update the config file.

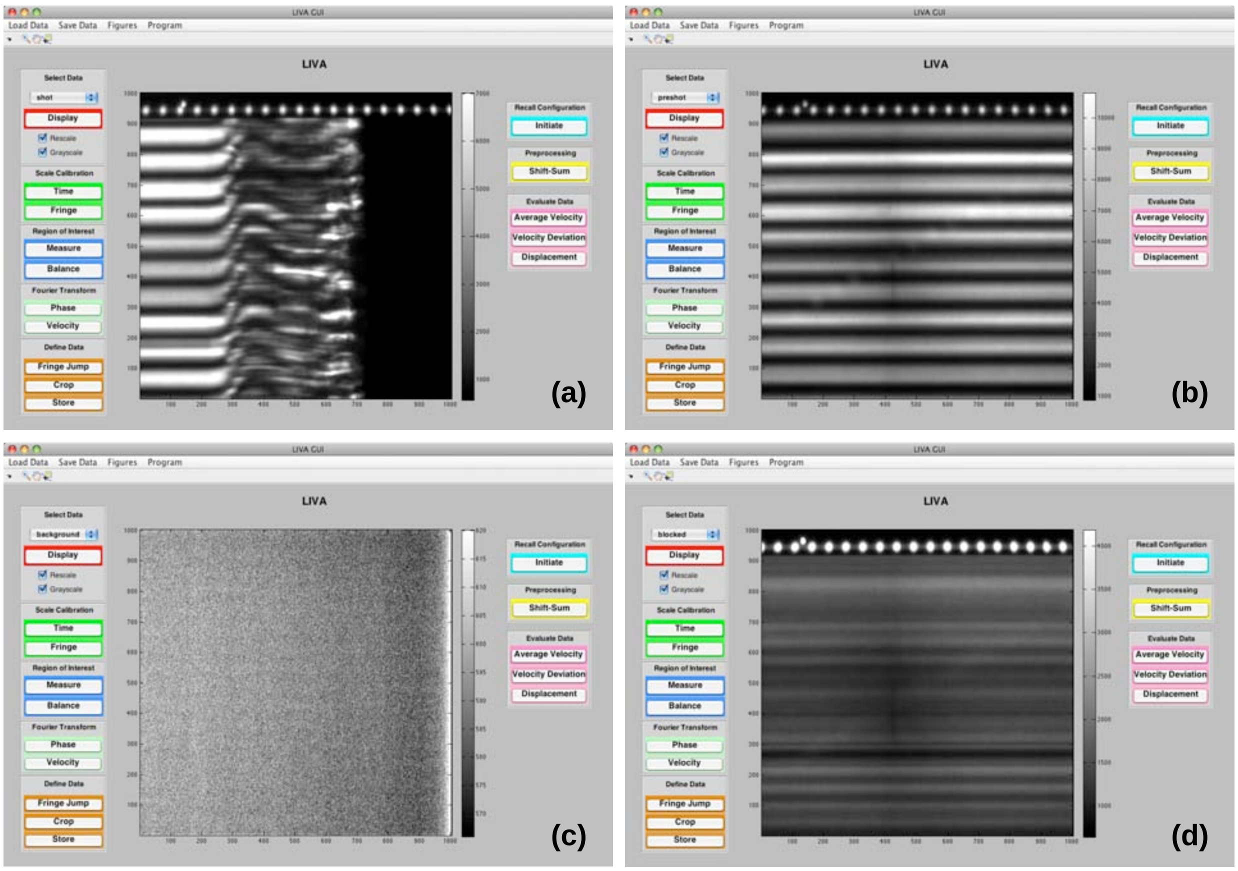

Figure 4.14. LIVA example 2: data images, (a) shot, (b) preshot, (c) background, and (d) blocked.

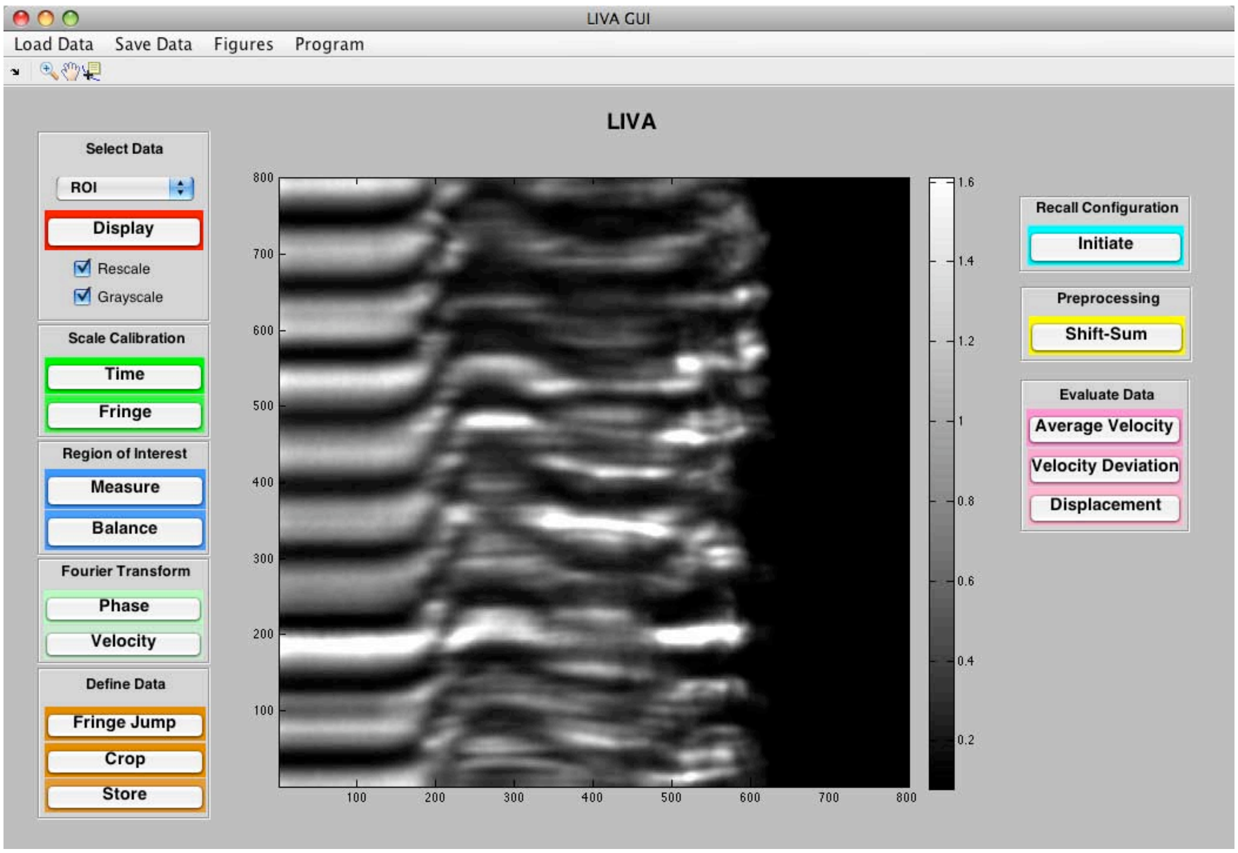

Figure 4.15. LIVA example 2: ROI of smeared data.

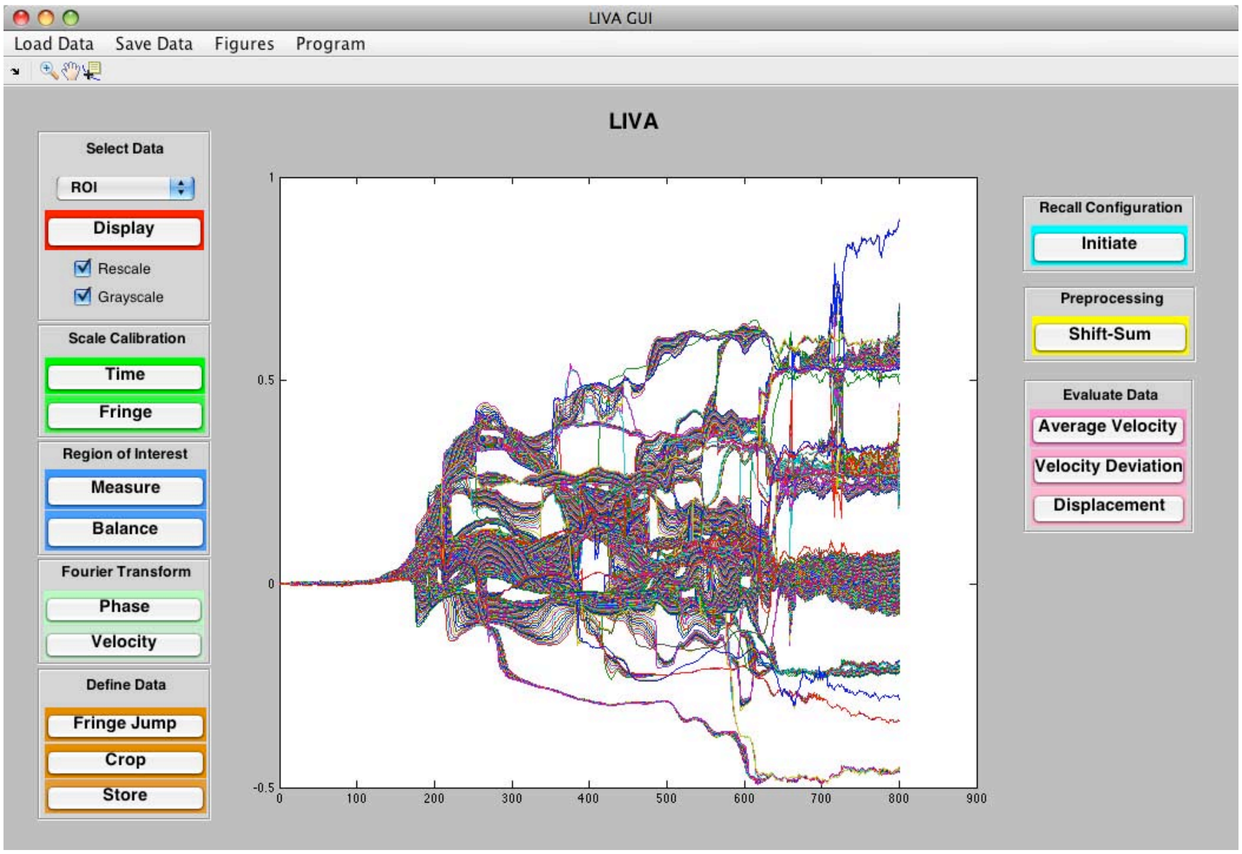

Figure 4.16. LIVA example 2: velocity profiles of smeared data.

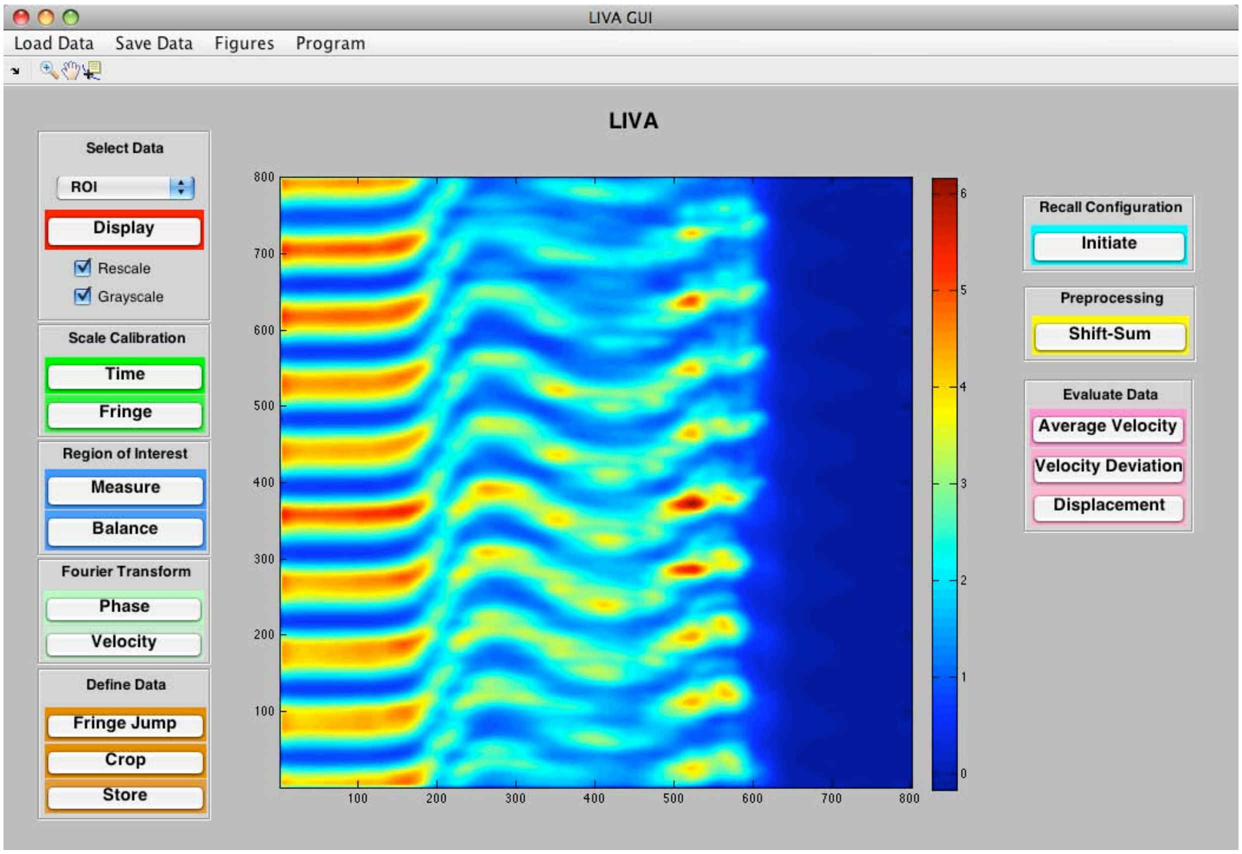

Figure 4.17. LIVA example 2: shift-sum image.

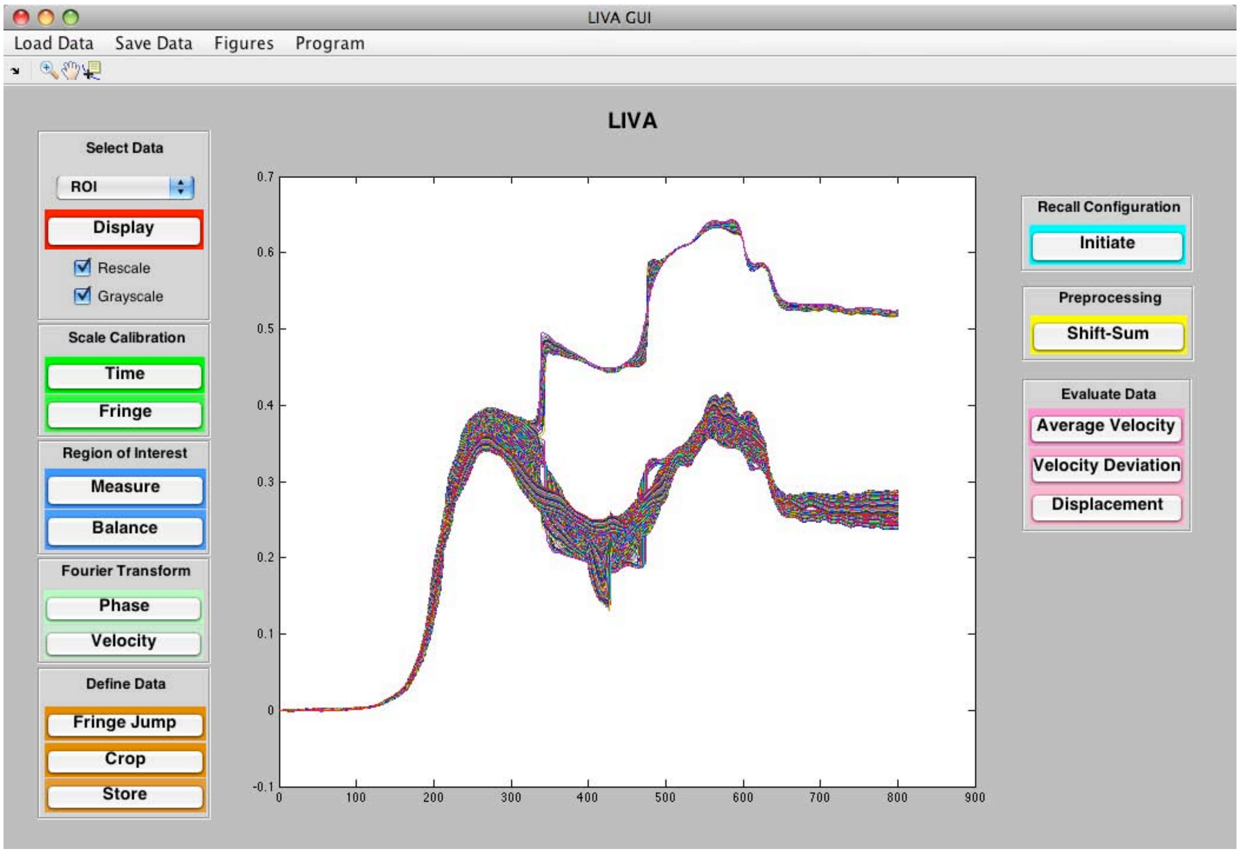

Figure 4.18. LIVA example 2: velocity profiles after shift-sum preprocessing.

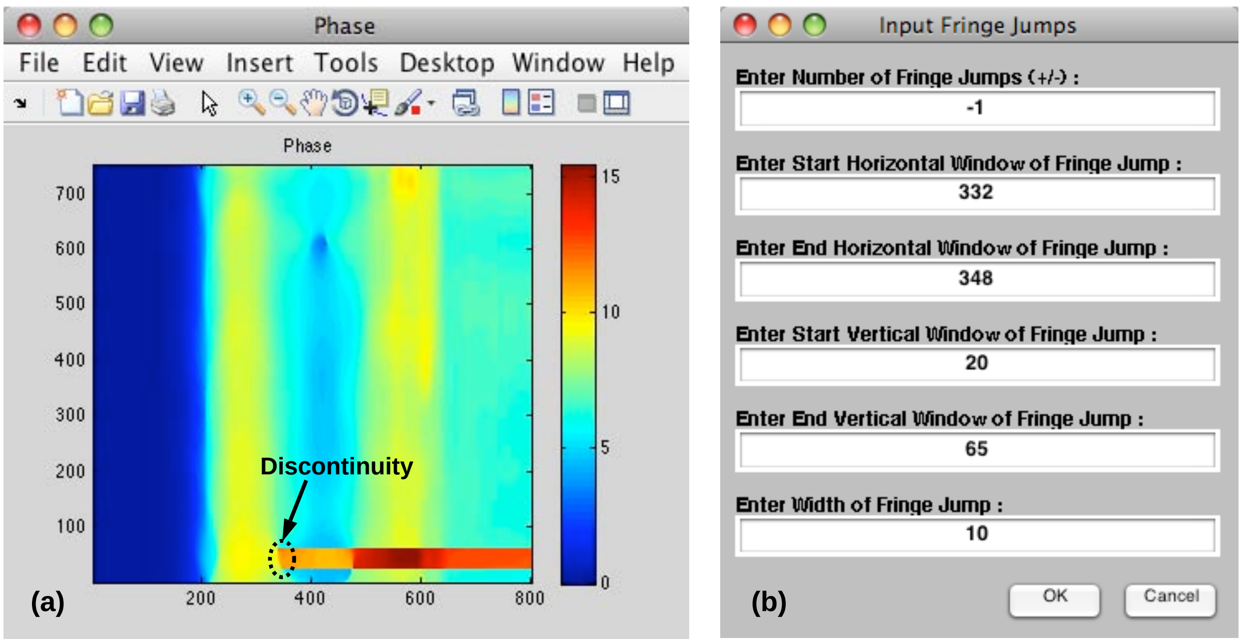

Figure 4.19. LIVA example 2: (a) phase of shift-sum data, and (b) window of fringe jump.

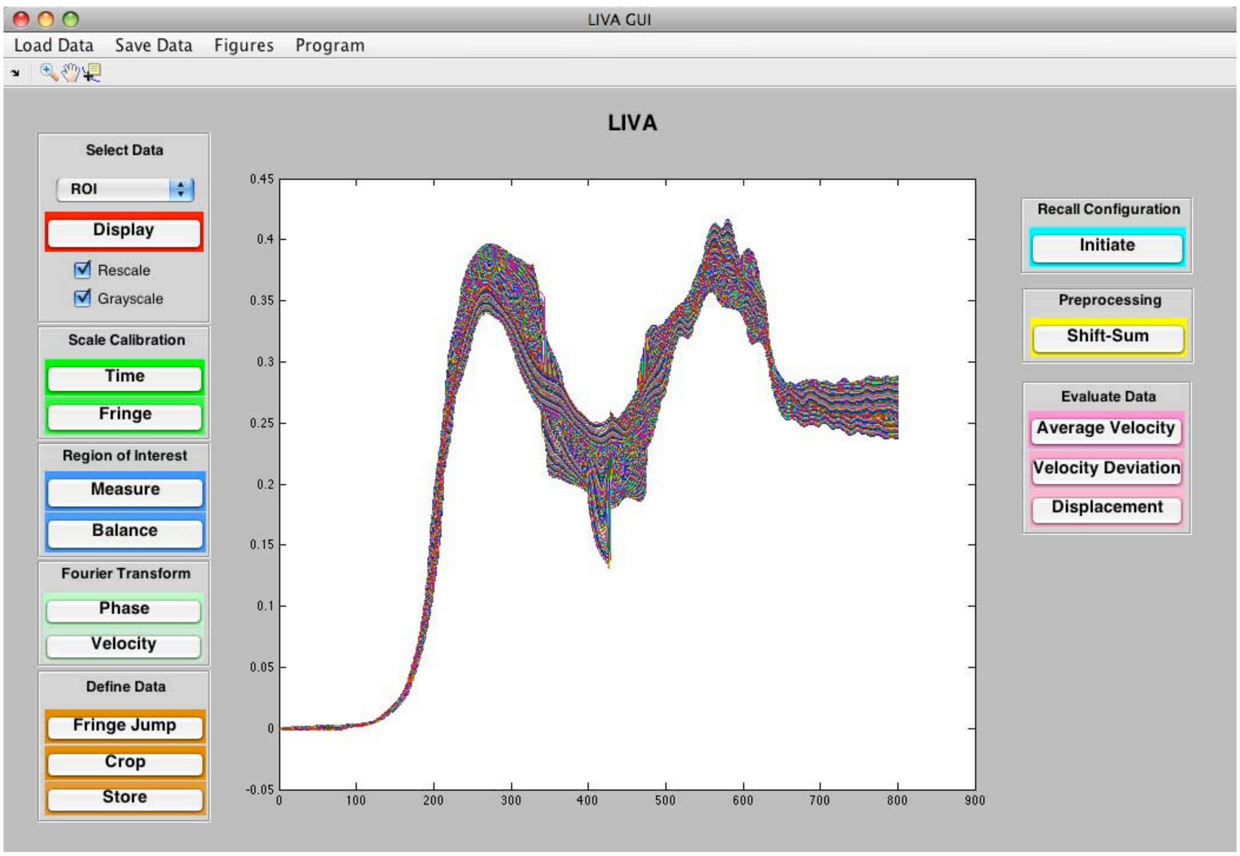

Figure 4.20. LIVA example 2: velocity profiles after fringe jump correction.

Chapter 5

Summary

5.1 Program features

The LIVA program accepts line-imaging ORVIS data from shock and isentropic compression experiments in the form of separate image files. The user crops the data image to the desired region of interest, and performs time and fringe calibrations. Heterogeneous signal levels within the data image may also be corrected using a balancing procedure. In the case of low image quality, a shift-sum algorithm may be applied to data in order to proceed with the analysis.

A Fourier transform method is employed to extract phase information from the interferogram of the ORVIS data. Spatially resolved velocity profiles are obtained by applying the experimental velocity-per-fringe to the phase map of the data. Discontinuities in the velocity profiles are removed by performing fringe jump corrections. Additional evaluations of the velocity profiles produce the spatially averaged velocity profile, the velocity deviation within the measured surface, and corresponding surface displacement.

Results generated by LIVA may be exported as text files for post-processing or be saved in various graphical formats. Finally, the parameters of the current analysis may be exported into a configuration file to be used in future LIVA analyses.

5.2 Future releases

Users should contact Tommy Ao (tao@sandia.gov) with questions, bug reports, and feature requests. Bug fixes will be made as necessary and will be distributed by email. No scheduled updates to LIVA is planned at this time, but new releases will be considered based on users' feedback.

References

[1] L. M. Barker and R. E. Hollenbach, "Laser interferometer for measuring high velocities of any reflecting surface," *Journal of Applied Physics*, vol. 43, pp. 4669–4675, November 1972.

[2] D. D. Bloomquist and S. A. Sheffield, "Optically recording interferometer for velocity measurements with subnanosecond resolution," *Journal of Applied Physics*, vol. 54, pp. 1717–1722, April 1983.

[3] K. Baumung, J. Singer, S. V. Razorenov, and A. V. Utkin, "Hydrodynamic proton beam-target interaction experiments using an improved line-imaging velocimeter," in *Shock Compression of Condensed Matter - 1996* (S.C. Schmidt and W.C. Tao, ed.), p. 1015, AIP Conf. Proc., 1996.

[4] W. M. Trott, M. R. Baer, J. N. Castaneda, L. C. Chhabildas, and J. R. Asay, "Investigation of the mesoscopic scale response of low-density pressings of granular sugar under impact," *Journal of Applied Physics*, vol. 101, p. 024917, January 2007.

[5] P. M. Celliers, G. W. Collins, L. B. D. Silva, D. M. Gold, and R. Cauble, "Accurate measurement of laser-driven shock trajectories with velocity interferometry," *Applied Physics Letters*, vol. 73, pp. 1320–1322, September 1998.

[6] C. A. Hall, J. R. Asay, M. D. Knudson, R. B. S. W. A. Stygar, T. D. Pointon, D. B. R. A. Toor, and R. C. Cauble, "Experimental configuration for isentropic compression of solids using pulsed magnetic loading," *Review of Scientific Instruments*, vol. 72, pp. 3587–3595, September 2001.

[7] T. Ao, J. R. Asay, S. Chantrenne, M. R. Baer, and C. A. Hall, "A compact strip-line pulsed power generator for isentropic compression experiments," *Review of Scientific Instruments*, vol. 79, p. 013903, 2008.

[8] T. Ao, J. R. Asay, S. Chantrenne, R. J. Hickman, M. D. Willis, A. W. Shay, S. A. Grine-Jones, C. A. Hall, and M. R. Baer, "The Veloce pulsed power generator for isentropic compression experiments," Sandia report SAND2007-8039, Sandia National Laboratories, December 2007.

[9] G. A. Mastin and D. C. Ghiglia, "Digital extraction of interference fringe contours," *Applied Optics*, vol. 24, p. 1727, June 1985.

[10] G. A. Fisk, G. A. Mastin, and S. A. Sheffield, "Digital image processing of velocity-interferometer data obtained from laser-driven shock experiments," *Journal of Applied Physics*, vol. 60, pp. 2266–2271, October 1986.

[11] M. Takeda, H. Ina, and S. Kobayashi, "Fourier-transform method of fringe-pattern analysis for computer-based topography and interferometry," *J. Optical Society of America*, vol. 72, pp. 156–160, January 1982.

[12] P. M. Celliers, D. K. Bradley, G. W. Collins, D. G. Hicks, T. R. Boehly, and W. J. Armstrong, "Line-imaging velocimeter for shock diagnostics at the OMEGA laser facility," *Review of Scientific Instruments*, vol. 75, pp. 4916–4929, November 2004.

[13] D. H. Dolan, "Foundations of VISAR analysis," Sandia report SAND2006-1950, Sandia National Laboratories, June 2006.

[14] W. J. Macy, "Two-dimensional fringe-pattern analysis," *Applied Optics*, vol. 22, p. 3898, December 1983.

[15] K. Nugent, "Interferogram analysis using an accurate fully automatic algorithm," *Applied Optics*, vol. 24, p. 3101, September 1985.

[16] C. Roddier and F. Roddier, "Interferogram analysis using Fourier transform technique," *Applied Optics*, vol. 26, pp. 1668–1673, May 1987.

DISTRIBUTION:

1	MS 0826	W. M. Trott, 1512
1	MS 0836	M. R. Baer, 1500
10	MS 1106	T. Ao, 1646
1	MS 1159	S. C. Jones, 1344
1	MS 1189	M. P. Desjarlais, 1640
1	MS 1189	T. A. Mehlhorn, 1640
1	MS 1195	J. R. Asay, 1646
1	MS 1195	C. S. Alexander, 1646
1	MS 1195	J.-P. Davis, 1646
1	MS 1195	D. H. Dolan, 1646
1	MS 1195	M. D. Furnish, 1646
1	MS 1195	C. A. Hall, 1646
1	MS 1195	R. J. Hickman, 1646
1	MS 1195	M. D. Knudson, 1646
1	MS 1195	J. Podsednik, 1646
1	MS 1195	S. Root, 1646
1	MS 1195	W. D. Reinhart, 1646
1	MS 1195	J. L. Wise, 1646
1	MS 9042	T. J. Vogler, 8246
1	MS 0899	Technical Library, 9536 (electronic copy)

Sandia National Laboratories