AUTOMATIC FRINGE ANALYSIS

Arnold Chiu, Ted Ladewski, and Jerry Turney

KMS Fusion, Inc.
700 KMS Place
PO Box 1567
Ann Arbor, MI 48106-1567
(313)769-8500 FAX:(313)769-1765

INTRODUCTION

To address the need for fast, accurate interferometric analytical tools, the Fringe Analysis Workstation (FAW) has been developed to analyze complex fringe image data easily and rapidly. FAW has been used for flow studies in aerodynamics and hydrodynamics experiments, and for target shell characterization in inertial confinement fusion research. This paper will describe three major components of the FAW system: input/output, fringe analysis/image processing, and visualization/graphical user interface.

Figure 1. FAW is a user-friendly system that facilitates automatic fringe analysis of interferometric image data.
The FAW Input/Output subsystem allows the operator to store and retrieve image data files to and from the disk. Such data files can be created via digitization of fringe images using either the fast, low-resolution video camera or a slow, high-resolution camera. FAW also provides software hooks for real-time, automatic control of interferometric image acquisition.

FRINGE ANALYSIS AND IMAGE PROCESSING

FAW extracts phase maps from digitized intensity images using any of three approaches: classical, phase-shifted, or carrier-injected. Light intensity, \( I(x,y) \), recorded at location \((x,y)\) in a fringe image is related to the underlying phase, \( \Phi(x,y) \), by:

\[
I(x,y) = I_o(x,y) * (1 + m(x,y) * \cos(\Phi(x,y) + \Phi_0))
\]

where \( I_o(x,y) \) is the illumination, \( m(x,y) \) is the local fringe contrast (e.g. surface reflectance or transparency), and \( \Phi_0 \) is the phase offset. Using the classical approach, FAW locates and traces fringe contours from a single, standard fringe image. It then interpolates the phase between these contours \[1,4\]. Although this approach needs only a single, standard fringe image, it cannot resolve inherent phase ambiguities without operator assistance.

In phase-shift interferometry \[6\], at least three, registered fringe images are made, each with a different reference beam phase \( \Phi_0 \). \( \Phi_0 \) is optically tuned to known values, \( \Phi_j \), to generate a set of phase-shifted fringe images \( I_j(x,y) \). The phase modulo \( 2\pi \) is then extracted directly using:

\[
\Phi(x,y) = \tan^{-1}(\sum_j I_j(x,y) \sin(\Phi_j) / \sum_j I_j(x,y) \cos(\Phi_j))
\]

FAW evaluates this function automatically and quickly at each pixel quickly using an arctangent lookup table.

In carrier-injection interferometry \[7\], one fringe image is made with a known tilt in the reference beam, thus introducing a spatial carrier frequency, \( \Phi_0 = \omega_0 * x \), into \( I(x,y) \). This allows \( \Phi(x,y) \) to be isolated in the Fourier domain, where \( I(x,y) \) separates into three main components (Fig. 2a, top). The phase modulo \( 2\pi \) is recovered by zeroing the middle and left peaks and then applying an inverse Fourier transform (Fig. 2a, bottom). Both the phase-shift and carrier-injection methods are highly accurate, do not suffer from ambiguities, and require no operator assistance. However, the carrier frequency introduced can mask important data.

Both the phase-shift and carrier-injected techniques recover the phase modulo \( 2\pi \), i.e. as a discontinuous phase map, \(|\Phi(x,y)|_{2\pi} \), with values limited to the range 0 to \( 2\pi \). This phase modulo \( 2\pi \) we will call "raw phase" \( \phi(x,y) \). FAW calculates the true phase \( \Phi(x,y) \) (Fig. 2b, bottom) by adding an offset function \( f(x,y) \) (Fig. 2b, middle) to the raw phase \( \phi(x,y) \) (Fig. 2b, top):

\[
\Phi(x,y) = |\Phi(x,y)|_{2\pi} + f(x,y)
\]
Figure 2. (a) Raw phase for carrier-injected images is extracted in the Fourier domain.
(b) True phase $\Phi(x, y)$ is recovered by adding an offset function $f(x, y)$ to the raw phase $\phi(x, y)$.

The most difficult part of calculating the true phase is in accurately determining the offset function $f(x, y)$. Due to image noise and poor image contrast, phase jump contours are often broken and the operator has to correct such discontinuity errors interactively. We present here two techniques for converting raw phase to true phase automatically instead.

Algorithm 1: Let the phase be $\Phi(\omega) = \Phi(\omega_x + i \omega_y).$ By definition, $\Phi(\omega) = \partial \Phi/\partial x + i \partial \Phi/\partial y.$ The derivative components can be estimated from the raw phase image $\phi(x, y) = |\Phi(x, y)|_{2\pi}$ by:

$$\partial \phi/\partial x = \min(\Delta_x \phi, \phi_{\text{max}} - \Delta_x \phi), \quad \partial \phi/\partial y = \min(\Delta_y \phi, \phi_{\text{max}} - \Delta_y \phi)$$  \hspace{1cm} (4)

where $\Delta_x \phi = \phi_{x+1} - \phi_x,$ $\Delta_y \phi = \phi_{y+1} - \phi_y,$ and $\phi_{\text{max}}$ is the maximum intensity value (e.g. 255 for a byte image).

Using a standard property of the Fourier transform $F$ [5]:

$$F(f'(\omega)) = i\omega F(f(\omega))$$  \hspace{1cm} (5)

the estimated true phase $\Phi(x, y)$ can be recovered by:

$$\Phi(x, y) = F^{-1}(F(\Phi(x, y)) / i\omega)$$  \hspace{1cm} (6)

where $\omega = \omega_x + i\omega_y,$
$F(\Phi(x, y)) = 0$ if $\omega = 0,$
and $F(\Phi(x, y)) = F(\Phi(x, y)) / i\omega$ if $\omega \neq 0.$
\( \Phi(x,y) \) is the estimated phase gradient extracted from the raw phase using (4). Algorithm 1 is computationally very efficient. However, application of the Fourier transform requires continuity of \( \Phi(x,y) \). Figure 3a shows the results for phase-shifted images of a smooth target shell that satisfies such requirements. Figure 3b shows the results for a NSWC carrier-injected image where errors due to discontinuities are quite apparent.

Algorithm 2: Discontinuity errors in the raw phase gradient are the main source of problems in constructing the offset function \( f(x,y) \). These errors occur at pixels with non-zero curl and may be found by checking the continuity assumption \( \partial(\partial \Phi / \partial y) / \partial x = \partial(\partial \Phi / \partial x) / \partial y \). FAW implements this check as a sum of the partials in the neighborhood around each pixel \( p_{i,j} \) (Figure 4a). If the sum is non-zero, a contour is broken and a discontinuity error is detected (Fig. 4b). All discontinuity errors in the image are masked out so that they can be safely ignored during true phase reconstruction. After assigning an known (arbitrary) phase to an initial point, a flooding algorithm is used to spread the integrated phase throughout the image. Results are shown in Figures 5 and 6.

![Figure 4. (a) Traverse neighbors of pixel \( p_{i,j} \) to ensure continuity in the local gradient. (b) If \( \Sigma_{i,j} \Phi(x,y) \neq 0 \), a phase jump contour is discontinuous.](image-url)
Figure 5. (a) Phase-shifted image set (b) Raw phase map (c) True phase map.

Figure 6. (a) Carrier-injected image (b) Raw phase map (c) True phase map.

VISUALIZATION AND GRAPHICAL USER INTERFACE

FAW is a DECwindows-based [2] system that provides a graphical user interface (GUI) for the intuitive visualization and manipulation of phase data. Since FAW has easily accessible functions and system and algorithm parameters, a user can learn how to use FAW in a matter of minutes. FAW can process image data to locate features, such as edges, or to visualize features with 2D contour and 3D surface plots.
A typical FAW session is shown in Figure 7. With a few mouse operations, the user has loaded an image set and has processed the set to extract phase information. The user can customize the system and hardware via the main window at the top of the screen. The computed raw phase image is displayed in the image workarea of the image window. A "toolbox" for editing the overlays is shown next to the image window. Among other tools are ones for creating magnified subimages, locating and tracking fringes across shock fronts, and examining the phase content in a pixel neighborhood or the cross-section of a shock stream. Functional and operational details on FAW are available in [3].

ACKNOWLEDGEMENT

The authors would like to thank KMS for its support, Bob Hays, Fred Schebor, and Jim Downward of KMS for their contributions to the FAW system, Bill Yanta of NSWC and Jim Trollinger of Metrolaser Inc. for their feedback on the initial prototypes. This work was supported by NASA NAS2-12531 and NSWC N6021-89-C-0057.

REFERENCES

1. Fringe Analysis System (Vol. 1-5), Ann Arbor, MI:KMS Fusion, Inc.
2. VMS DECwindows Programming (Vol. 1-2), Maynard, MA:DEC.