Adaptive DFT-Based Interferometer Fringe Tracking

Edward Wilson

Intellization, 454 Barkentine Lane, Redwood Shores, CA 94065, USA
Email: ed.wilson@intellization.com

Ettore Pedretti

Harvard-Smithsonian Center for Astrophysics, 60 Garden Street, Cambridge, MA 02138, USA

Astronomy Department, University of Michigan, 914 Dennison Building, Ann Arbor, MI 48109, USA
Email: epedrett@umich.edu

Jesse Bregman

NASA Ames Research Center, Mail Stop 269-1, Moffett Field, CA 94035, USA
Email: jesse.bregman-1@nasa.gov

Robert W. Mah

NASA Ames Research Center, Mail Stop 269-1, Moffett Field, CA 94035, USA
Email: robert.w.mah@nasa.gov

Wesley A. Traub

Harvard-Smithsonian Center for Astrophysics, 60 Garden Street, Cambridge, MA 02138, USA
Email: wtraub@cfa.harvard.edu

An automatic interferometer fringe tracking system has been developed, implemented, and tested at the Infrared Optical Telescope Array (IOTA) Observatory at Mount Hopkins, Arizona. The system can minimize the optical path differences (OPDs) for all three baselines of the Michelson stellar interferometer at IOTA. Based on sliding window discrete Fourier-transform (DFT) calculations that were optimized for computational efficiency and robustness to atmospheric disturbances, the algorithm has also been tested extensively on offline data. Implemented in ANSI C on the 266 MHz PowerPC processor running the VxWorks real-time operating system, the algorithm runs in approximately 2.0 milliseconds per scan (including all three interferograms), using the science camera and piezo scanners to measure and correct the OPDs. The adaptive DFT-based tracking algorithm should be applicable to other systems where there is a need to detect or track a signal with an approximately constant-frequency carrier pulse. One example of such an application might be to the field of thin-film measurement by ellipsometry, using a broadband light source and a Fourier-transform spectrometer to detect the resulting fringe patterns.

Keywords and phrases: fringe tracking, DFT, interferometry, IOTA, real time.

1. INTRODUCTION

The infrared-optical telescope array (IOTA), shown in Figure 1, is a 3-aperture-long baseline Michelson stellar interferometer located on Mount Hopkins near Tucson, Arizona. Three 45 cm collectors can be located along a 15 m by 35 m L-shaped array, supplying visible and near-IR light to pupil-plane beam combiners. The operational details and scientific accomplishments of IOTA have been well documented in [5, 6] and at http://cfa-www.harvard.edu/cfa/ oir/IOTA.

This paper reports on the development of an algorithm designed and used to simultaneously minimize the optical path differences (OPDs) for the three baselines (A-B, A-C, and B-C) provided by IOTA's three apertures.

1.1. Fringe tracking goals

Details of the relevant interferometric derivations are covered thoroughly in other references such as [1, 2]. From a signal processing perspective, it is important to know that the governing physics of stellar pupil-plane interferometry result in an ideal signal that looks like that shown in Figure 2.

The idealized fringe packet function is a sinc function multiplied by a sinusoid, and can be represented with (1),

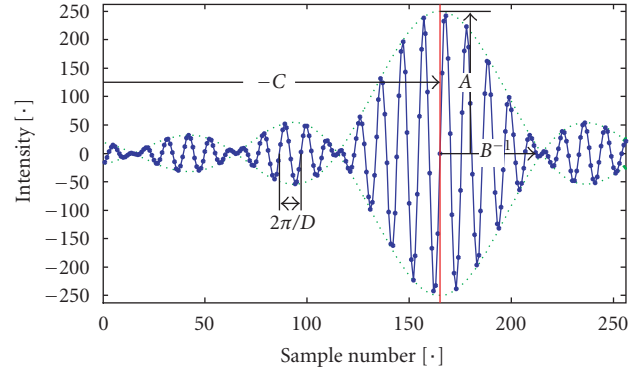FIGURE 1: Infrared-optical telescope array (IOTA).

FIGURE 2: Idealized interferogram, sinc-function envelopes, and center shown. (In this paper, the $[\cdot]$ in the figure axis labels indicates that the variables are unitless.)

where y is the mean-subtracted intensity,[1] x is the sample number, and A, B, C, D, E are parameters defining, respectively, the amplitude, sinc-function width, sinc function center, sinusoid (fringe) frequency, and sinusoid phase shift:

$$y = A \operatorname{sinc}\left(B(x + C)\right) \sin\left(D(x + E)\right). \tag{1}$$

Many equivalent variants on this functional form are of course possible (e.g., substituting $\cos(dx - e)$ for $\sin(D(x + E))$); this one was chosen to facilitate the gradient-based optimization procedure described in [3, 4]. The sinusoid in this function comes from the interferometric combination of light from two apertures (a telescope pair), and the sinc function enters due to the Fourier transform of the instrument's spectral response (which is uniform over a fixed range). A is related to fringe visibility (or contrast) and varies from object to object. Visibility is the most important measured quantity because it is related to the object brightness distribution. B depends on the filter used and on the composition and temperature of the object measured. C reflects the optical path difference (OPD). D, the fringe frequency, is related to the central wavelength of the light passing through the filter and to the length of the scan. Although E is not an independent variable in theory, in practice, dispersion, noise, atmospheric disturbances, and the object itself cause it to vary from this idealized case. The relative shift between the fringes obtained from three baselines, or "phase closure" enables partial retrieval of this information and, consequently, the possibility of obtaining high-resolution images of distant astronomical sources.

The center of the fringe packet, or interferogram, corresponds to the point at which the OPD between each of the two collectors and the source (star) is zero. The path lengths are adjusted with slow- and fast-moving mirrors, driven to account for slow (and well calculable) effects such as earth ro-

tation, as well as fast effects such as atmospheric turbulence. Ideally, with perfect compensation and no atmospheric distortion, the fringe packet would be fixed in the center of the scan. The interferogram represents the samples taken as a piezo-driven mirror is driven through a stroke of typical length 25 microns over a period of typically 333 milliseconds. If the center of this mirror scan stroke is not sufficiently close to the true OPD zero point, the fringe packet will be lost from view, and no science data will be available.

In practice, an idealized sinc function such as this is not seen. The band edges are often obscured by noise. However, the fringe tracking algorithm was designed to work on actual data, so it is very robust to significant deviations from this idealized form.

Thus the goal of an interferometer fringe tracker is to analyze incoming interferograms and provide on-line adjustments to the piezo-scanning delay-line mirror to keep the fringe packet centered within the scan window. It should be as follows.

(1) Robust to noise and anomalies in the data: absolute accuracy is not as important as keeping the fringe within the scan window.

(2) Require few if any manual adjustments: autonomous adaptability is needed to cover widely varying seeing conditions and object intensities.

(3) Computationally efficient: a minimal amount of computation time is available due to the limited resources and need for fast scanning, typically 3 Hz.

Maximum accuracy is less important than robustness, since as long as the fringe packet is in the scan window, it can be analyzed in postprocessing. As fringe tracking accuracy increases, however, it becomes possible to reduce the stroke length of the piezo scanner, thus increasing the overall rate of data collection. There is a secondary benefit in reducing stroke length in that, for a constant scan velocity, a shorter stroke will mean less time between scans, which reduces the time during which the atmosphere may have changed, reducing the average size of the fringe packet random motion.

---

[1]For the IOTA detector, raw data from each channel is divided by the mean across the full scan. Then the complementary channels are subtracted. Normalization by the sum of the complementary channels is performed for data analysis, but not for fringe tracking. However, fringe tracking algorithm performance is independent of the specifics of this mean-subtraction and normalization process.
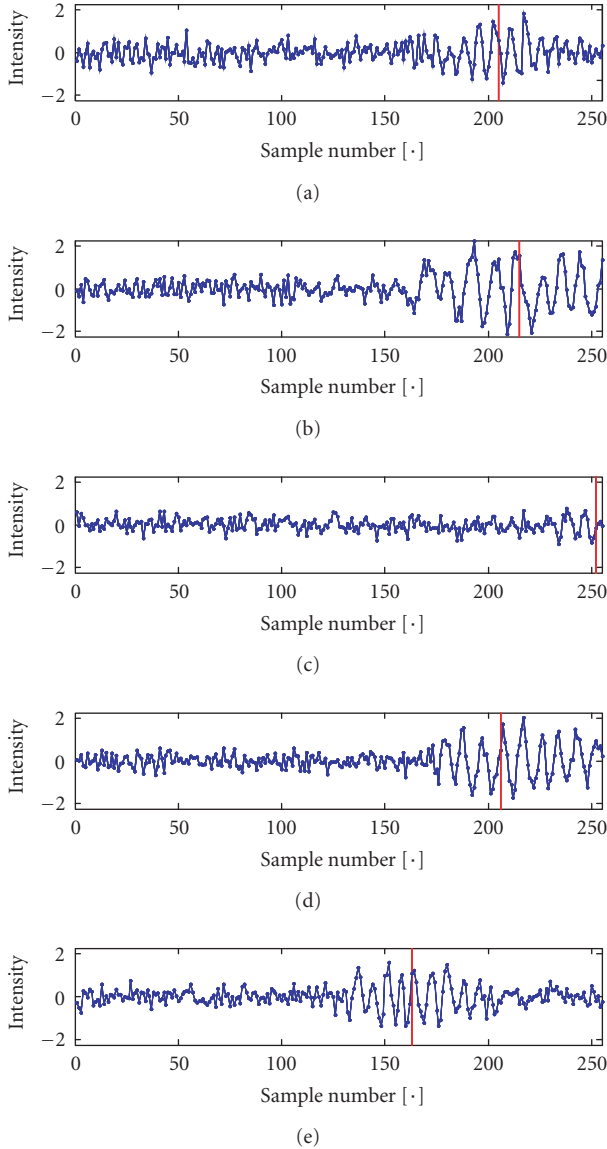
(a)

(b)

(c)

(d)

(e)

FIGURE 3: Typical sequence of scans ((a) scan no. 12, confidence = 4.6; (b) scan no. 13, confidence = 6.1; (c) scan no. 14, confidence = 2.2; (d) scan no. 15, confidence = 7.0, and (e) scan no. 16, confidence = 6.4).

Figure 3 shows a typical sequence of scans from IOTA. This data was taken on April 20, 2004, on the B-C telescope pair, targeting star HD126035, with the fringe tracker turned off. These are 5 consecutive 256-point scans out of the 200-scan data set, taken at a scan rate of 3 Hz, with a 25-micron scan length. The IONIC beam combiner works in H band which translates to a wavelength of 1.65 microns. The H band filter used has a bandwidth of $0.35\,\mu$m. Depending on the scan length and number of samples per scan, sampling can range from about 4–10 samples per fringe.

Although the idealized form of the sinc-sinusoid from (1) can be seen, there is significant background noise, variability of the sinusoid (fringe) frequency from scan to scan and within each scan, and the packet center can be seen to move

randomly between one scan and the next. Although these interferograms appear to have relatively consistent quality (with the exception of scan #14, which has drifted almost completely out of the window), it is not uncommon to have significant changes in quality (noise level, jump size, fringe clarity) from one scan to the next.

Shown along with the raw data, the solid vertical lines indicate the identified fringe packet centers that could have been used by the fringe tracking software to recenter the piezo scan, had fringe tracking been turned on. The relative confidence in the fringe-center identification is indicated in the title of each subplot and will be discussed later.

Significant sources of noise include atmospheric turbulence, vibration, photon noise, and detector noise. The goal of the fringe tracking system is to perform coherencing (versus cophasing) by controlling the OPD to allow the interferogram to be captured in the presence of bad seeing conditions and fainter objects. The controller works by identifying the fringe-center locations on all 3 interferograms following each scan and then adjusting the centers of travel of the piezo-driven scanning mirrors, attempting to keep the fringe packets centered in all 3 scan windows. The computing and actuation aspects of the control system are described by Pedretti and Traub [2, 6]; the present article details the fringe tracking algorithm and aspects of its software implementation.

Due to the noise sources present and the lack of a sufficiently representative simulation, the fringe tracking algorithms presented here were developed through extensive testing on actual data sets from IOTA, dating back to 1997 (as opposed to working with simulated data).

1.2. Related research

Observations performed with long-baseline ground-based optical/infrared interferometers are strongly affected by the turbulent atmosphere. Turbulence can reduce the visibility of fringes in many ways as described in [7] for pupil-plane (or coaxial) beam combination and in [8] for image-plane beam combination. Turbulence randomly modulates the phases of the fringes which can then become unusable for image reconstruction. Using three or more telescopes enables reduction of this atmospheric phase contamination. This is done through the closure-phase technique pioneered in radio astronomy by [9] and recently applied to long-baseline optical interferometry [10] allowing the first image of an astronomical source (the binary star Capella) to be obtained by an optical interferometer. More recently, optical and infrared interferometry has been able to provide information on the morphology of stellar sources [11] and extragalactic sources [12].

The necessary condition for obtaining meaningful closure-phases is that the three fringe packets must all be present in the same temporal interval. This is achieved by keeping the optical path difference (OPD) to a minimum.

Fringe tracking was used in interferometry since the very beginning of the field, when Michelson and Pease [13] used a prism for dispersing and acquiring fringes visually at the

20-foot interferometer. Labeyrie [14] used the same system and demonstrated fringe acquisition on a two-telescope interferometer. Several systems have been proposed since then, for correcting the optical path [15, 16]. GDT (also called dispersed fringe tracking when applied to image plane interferometry) has been routinely used at several interferometric facilities [17, 18, 19].

When IOTA relied on a single baseline, the fringes were usually kept inside the scan interval manually by the observers. The installation of the third telescope at IOTA required an increase in the level of automation in the instrument, because manual tracking is not practical with three baselines to adjust. In particular, the requirement to measure closure-phases necessitated a system capable of keeping the fringe packets in the center of the scan using the existing hardware dedicated to acquiring science data. Fringes must be acquired in the same coherence time in order to measure a closure-phase. A coherencing algorithm is very useful to localize the position of a fringe packet and correct the OPD in order to compensate metrology errors and atmospherically induced fluctuations in the optical path. This maximizes the superposition of the fringe packets and the signal-to-noise-ratio (SNR) of the closure-phase signal when this is averaged in the complex plane as shown in [20].

The remainder of this section summarizes fringe tracking developments at IOTA. Wilson developed a method, summarized in this paper, that used the envelope of the interferogram to identify the packet center, and a gradient-based optimization method for refinement of this estimate [3]. Although fast and robust, it did not make use of the fringe frequency, leading to the present research which makes this improvement. This was an offline study using IOTA data taken in 1997.

Morel and others in the IOTA team worked to implement the core aspect of Wilson's 1999 algorithm on the IOTA scanning hardware [21]. The fringe-center identification aspect of the system was found to be very robust and accurate even with very noisy signals, but the slow response of the control communications and actuation hardware made the overall control system ineffective. The control computing, communications, and actuation hardware was subsequently upgraded to permit further implementation efforts [6].

Pedretti developed a fringe tracking algorithm taking a completely different approach, based on double Fourier interferometry (DFI) [2, 22]. This method calculates the group delay of fringes dispersed with DFI, which is used to obtain the wavelength-dependent phase from the fringe packet. This method has also been implemented at IOTA on the current hardware, and is used there regularly. A performance comparison of the different approaches at IOTA is presently underway.

Thureau developed a fringe envelope tracking algorithm at COAST, which was subsequently implemented for testing at IOTA [23].

Gradient-based optimization, motion prediction, and other offline analyses are discussed in [4]. As compared to that publication, the present article uses data from 2004 and focuses on the adaptive DFT-based tracking algorithm, whereas [4] focuses on the IOTA implementation issues, envelope-based tracking, and offline gradient-based optimization of all packet parameters.

1.3. Approach

Guided by a background in signal processing and system identification (ID), the original approach taken towards fringe tracking was to fit the parameters in (1) to the data on each scan, with the fringe center then contained in C. A nonlinear, gradient-based optimization was developed to perform this, with extensive testing and tuning on representative IOTA data sets from 1997. This nonlinear optimization required a reasonably close initial estimate for C, which was provided by processing the fringe packet envelope. As it turned out, the accuracy of this initial estimate was generally within a sample or two (out of 256 points in a scan, typically) of the result following the full nonlinear ID. Given implementation constraints and the existence of other more significant error sources, it was decided that this initial estimate processing could serve as the online fringe ID algorithm. This was tested online in 1999 and 2000 [21].

In 2002, following the instrument control hardware upgrades and in preparation for a second implementation attempt, the algorithm was updated. The original envelope-based algorithm basically drew an envelope around the data and found the hump, thereby completely ignoring the fringe frequency, D. As can be seen in the example data given previously, the fringe frequency is visible in the fringe packet, and is relatively obscured by noise outside the center (due to the smaller envelope). The improvement looks for intensity amplitude at the fringe frequency, rather than at all frequencies (as the envelope-based ID did). See, for example, scan #14 in Figure 3. In that case, the envelope would not be a clear signal, but focusing on the expected fringe frequency leads to an accurate identification even with very little of the fringe packet in the window. This is accomplished with an efficiently implemented sliding window discrete Fourier transform (DFT). This updated algorithm was implemented in February 2002 at IOTA, with testing on simulated fringes through the instrument, and later on-the-sky testing with all 3 apertures in May 2002. Being more physically based, the change was made with the expectation that it would be more robust for future data and algorithm changes.

2. DFT-BASED TRACKING

The algorithm is summarized in Section 2.1, and then the individual steps are outlined in subsequent sections.

2.1. DFT-based tracking algorithm summary

(1) A window (nominally of a length containing two fringe periods, but can be set to any integer) is passed over the data, where a single-frequency discrete Fourier transform (DFT) is calculated to try to detect the expected fringe frequency (this frequency is adaptively updated—by changing the window size—after each scan). The DFT is calculated 5 times for each

scan, using window sizes of nominal plus $[-4, -2, 0, +2, +4]$. For example, if the nominal size is a 17-sample window covering 2 cycles, then the DFT is calculated for 13-, 15-, 17-, 19-, and 21-point windows. The number of points in the window is odd, so the center lands on a point. The magnitudes (i.e., the root sum of squares of the real and imaginary parts is taken, although that may not be essential) of these DFT results are used to determine the nominal window size for the next scan.

(2) Each of the 5 DFT results is smoothed using a rectangular averaging filter.

(3) The point-by-point maximum of the 5 smoothed DFT results, referred to as the composite DFT magnitude, is taken for further processing. Steps (1), (2), and (3) make this result more robust to intra-packet fringe frequency variations than a single-frequency DFT scan would be. The frequency corresponding to the largest DFT magnitude is chosen as the nominal frequency for the following scan—providing adaptive response to changing interferogram properties, and eliminating the need to initially set this carefully.

(4) A fringe-packet-finding template is convolved with the composite DFT magnitude, providing a peak when the composite DFT magnitude matches the template shape. For computational efficiency, a rectangular template is used in place of a sinc-shaped template.

(5) The sample corresponding to the maximum value of the previous step is used as the identified fringe-packet center.

(6) A confidence metric is calculated based on the relative magnitudes of the composite DFT magnitude near the ID'ed center and the background.

(7) The previous steps are performed on all aperture pairs (3 in the case of IOTA), and the ID results and corresponding confidence metrics are combined to determine the scan centers for the next scan (to begin within a couple of milliseconds).

2.2. Example data

The algorithm steps are presented in detail using actual data, as shown in the following figures. They were generated using data collected from the BC-fringe (from apertures B and C) of the 16th scan of the IOTA-25 dataset on April 20, 2004; targeting star HD126035; RA (J2000): 14.390278; Dec (J2000): -11.713889. This interferogram is also shown in Figure 3. Figure 4 shows the normalized raw data (the complementary pair, B-C), as well as the result of the center identification that came after all steps were completed.

2.3. DFT calculation

Figure 5 shows how the DFT window—in this case a 17-point window nominally containing two fringe wavelengths—is passed over the raw data. The purpose of the DFT is to locate areas in the scan where the expected fringe frequency is present. A few things are done to greatly improve the efficiency of the DFT calculation—note it is not calculated
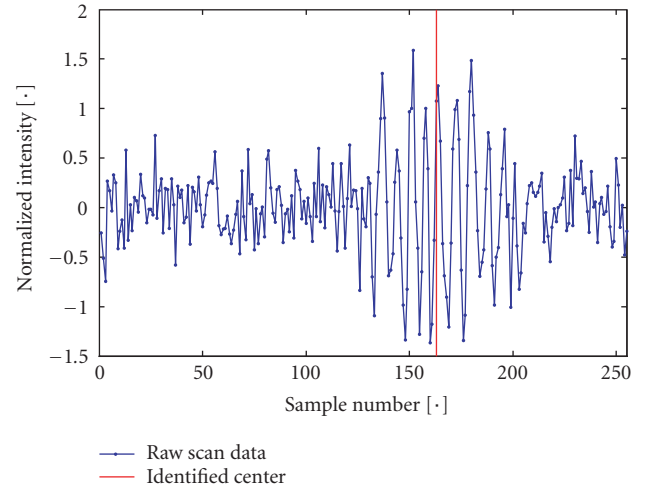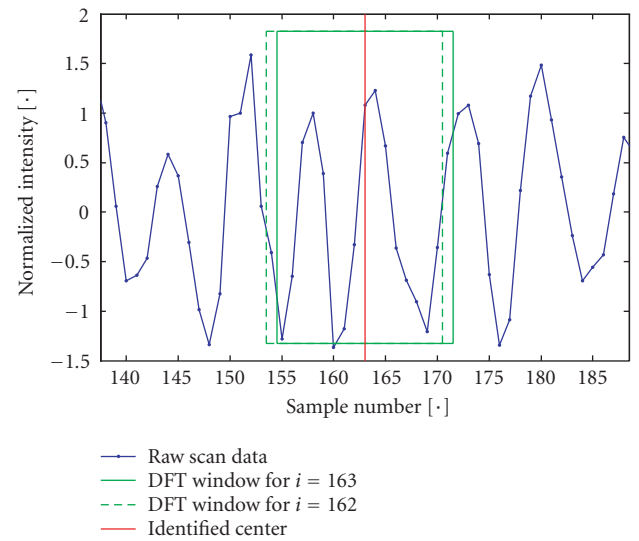

Figure 4: Normalized raw data.

Figure 5: DFT sliding window.

as an FFT. This DFT calculates the magnitude of the signal in only one frequency bin—that nominally corresponding to the fringe frequency. Also, a rectangular window is used, which enables very fast computation as the window is passed over the data. Calculating each new data point requires adding a term for the incoming sample and subtracting a term for the leaving sample. So, for example, to calculate the DFT for the fringe frequency centered at sample #163 in the figure uses the DFT result for sample #162, then adds a term for point #171 ($= 163 + (17 - 1)/2$, the new point in the sliding window), and subtracts a term for point #154 ($= 162 - (17 - 1)/2$, the point dropping out of the sliding window).

While it is efficient, *this implementation of the DFT calculation is also exact*. By doing this calculation only for the frequency component of interest, this is faster than an FFT. FFT compute time is proportional to $N \log_2 N$, where N is the

window length. A full-spectrum DFT would be N^2, whereas this single frequency DFT compute time is proportional to N. But a second and far more significant level of optimization is achieved by using a rectangular window and sliding it across the scan, one sample at a time, as follows.

(1) For implementation in C, the real and imaginary parts are calculated separately. So sine and cosine functions corresponding to the expected fringe frequency (as defined by the window size and bin number) are calculated as, for example,

$$\cos\left(\frac{2\pi(b-1)}{N}k\right), \qquad \sin\left(\frac{-2\pi(b-1)}{N}k\right), \qquad (2)$$

where b is the bin number (e.g., 3 for a window nominally containing two wavelengths), N is the window length, and k ranges from 0 to 255 for a 256-point scan. To further improve run-time performance, these sine and cosine operations could be made once on startup only and storing the 256-long vectors corresponding to each potential window size to be considered (e.g., $7, 9, 11, \ldots, 31$).

(2) The scan intensity vector, y, is multiplied point by point with these sine and cosine vectors—once per scan, resulting in 256-long vectors, Y_{real} and Y_{imag},

$$
\begin{aligned}
Y_{\text{real}}(k) &= y(k)\cos\left(\frac{2\pi(b-1)}{N}k\right), \\
Y_{\text{imag}}(k) &= y(k)\sin\left(\frac{-2\pi(b-1)}{N}k\right).
\end{aligned}
\qquad (3)
$$

(3) The real and imaginary parts of the DFT result for *only* the first sample in the scan (e.g., $X_{\text{real}}(N_2)$, where $N_2 = (N-1)/2$) are calculated by summing the above-calculated real and imaginary vectors over a window of length N,[2] for example,

$$
\begin{aligned}
X_{\text{real}}(N_2) &= \sum_{i=-N_2}^{N_2} Y_{\text{real}}(N_2 + i), \\
X_{\text{imag}}(N_2) &= \sum_{i=-N_2}^{N_2} Y_{\text{imag}}(N_2 + i).
\end{aligned}
\qquad (4)
$$

(4) Then, to calculate the real and imaginary DFT results at successive samples throughout the middle of the scan, the calculation is just one add and one subtract for the real and imaginary parts, corresponding to points entering and leaving the window. In particular, this is possible because a rectangular window is used. This computational optimization will change the phase of the result (and therefore the real and imagi-

nary parts), but the magnitude is still exact. The phase imparted by this step may easily and exactly be removed if phase information were important. So

$$X_{\text{real}}(k) = X_{\text{real}}(k-1) + Y_{\text{real}}(k+N_2) - Y_{\text{real}}(k-N_2-1). \quad (5)$$

(5) Since only the magnitude (versus the phase) is used, the real and imaginary results are combined accordingly (the square root could probably be omitted to improve speed if needed). Division by the window length also occurs at this point, to enable meaningful comparison between different DFT results.

(6) All computations are done using floats (versus doubles), since this provides sufficient accuracy.

(7) The DFT calculation does not depend at all on whether the window or scan size is a power of 2 (as the FFT does to some extent—although the non-power-of-two inefficiency is very slight for some FFT implementations such as FFTW [24]).

The DFT window size is chosen in this case to nominally contain exactly two fringe wavelengths. The algorithm requires the window size to be an odd integer whose selection is discussed later. Because of the way the DFT-calculation algorithm has been implemented, window sizes of 3, 4, 5, and so forth wavelengths could be calculated without changing the computation time. However, for larger windows, there is a possibility of the frequency changing during the window, which would distort the DFT calculation (i.e., when the time corresponding to the window size is less than the coherence time). Two wavelengths appears to be a good compromise between accuracy on clean scans (with higher coherence time, more accuracy would be possible with a larger window, but on clean scans tracking accuracy is not difficult) and noisy scans (if the coherence time is much below two wavelengths, there are probably no fringes to be seen).

For calculation of the DFT over the full spectrum, the first component ("bin") would correspond to the average value, the second component would correspond to a full wavelength extending across the full window, and the third component would correspond to two full wavelengths. Since the window size was chosen to cover two full fringe wavelengths, we calculate only the third component of the full-spectrum DFT. The real and imaginary parts are computed and then combined to produce the magnitude. The phase information is not used. This result is then smoothed using a sliding window having the same width as the DFT window (two wavelengths in this case), with the mean over the window producing the result shown in Figure 6. This step is computationally efficient, and reduces the variability in the DFT results.

Even though it is calculated very differently, the result is very similar to that resulting from the envelope-finding calculations described in [3]. While the envelope-finding calculations provided excellent results, this DFT calculation is more physically based and is expected to be more robust for noisy signals.

---

[2]The details of handling the beginning and ending of the scan are not covered here, but of course must be addressed.
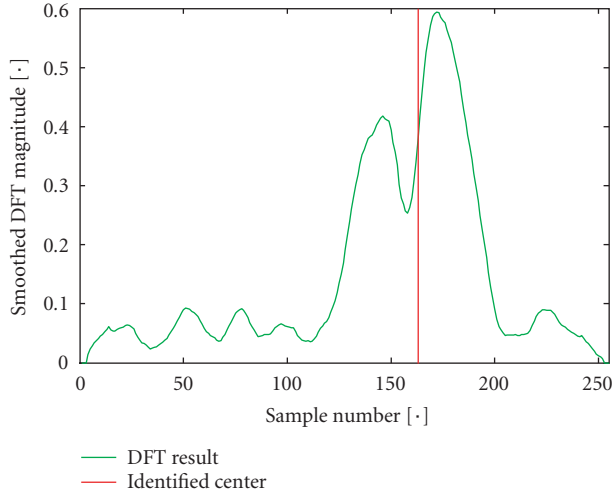
FIGURE 6: Smoothed DFT magnitude, using, in this case, a 17-point sliding window.

FIGURE 7: Smoothed DFT results for 13-, 15-, 17-, 19-, and 21-point windows.

The DFT can be thought of as a sampled version (exists only at the bin frequencies) of the discrete-time Fourier transform (DTFT) of the continuous signal convolved with the DTFT of the window function (in the simplest case, as we have here, this is a uniform window of finite length). With an infinitely wide window, the DTFT of the window would be an impulse, so the convolution would not distort the DTFT of the signal. With a finite window, two effects occur.

(1) *Reduced resolution*: unlike an impulse, the mainlobe of the window function has some finite width. Convolving this with the signal DTFT may make it impossible to resolve between two frequency components.

(2) *Leakage*: the component at one frequency leaks into that at another component due to the spectral smearing.

If the goal is to measure the DTFT of the signal, then ideally one would like a window with a DTFT of a thin mainlobe and small sidelobes, but usually there is a tradeoff between the two. A rectangular window has a relatively narrow mainlobe, while Hanning, and so forth, windows have a wider mainlobe (worse), but have smaller sidelobes (better). No matter the shape of the window, the mainlobe gets narrower as the number of points in the DFT increases. For nonstationary signals as we have here, at some point you do not want to increase the window size because the frequency content is changing. In this case, the length and shape of the window are chosen so that the Fourier transform of the window is narrow in frequency compared with changes in the FT of the signal [25].

However, in this application, the main concerns are not with resolution or leakage since our target, the fringe frequency, is changing from scan to scan and within a scan. Reduced resolution actually helps insulate the result from these variations. By adapting to these fringe frequency changes, the DFT result is useful for fringe tracking.
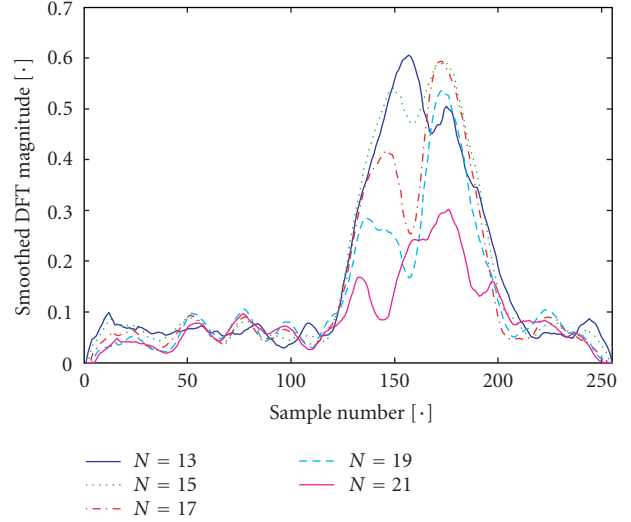
2.4. Combination of DFT results at multiple frequencies

Figure 7 summarizes the results of the smoothed DFT calculations using five different window sizes. Since the fringe frequency is not known exactly, and may change, this algorithm adapts to use the best window size. For every scan, the smoothed DFT calculations discussed above run five times: once for the window size used on the previous scan (e.g, 17 samples); once each for 2 and 4 samples larger; and once each for 2 and 4 smaller. The window size (N, as shown in the plot legend, is the number of samples) corresponding to the maximum smoothed DFT value is chosen and carried through to the next scan as the nominal window size. To prevent N from increasing or decreasing too rapidly during periods of low signal-to-noise ratio, only one step up or down is permitted per scan. Also, to prevent unnecessary changes, the nominal N is changed only if the maximum exceeds that of the nominal by 5%. So in the example shown, where the highest DFT result occurs for $N = 13$ at sample #157, it was within that threshold, so the nominal N carried forward was 17.

The "composite DFT magnitude" is formed by taking a point-by-point maximum over the 5 smoothed DFT results. This is the vector that is passed on for further processing. As can be seen by the 5 individual curves here, the composite DFT magnitude is more representative of the fringe packet than any single smoothed DFT result would be. This is due to the intra-packet variations in fringe frequency, caused primarily by atmospheric distortion. Since it uses 5 different frequencies, this composite DFT magnitude does not have a physical meaning, although it efficiently captures the essential information needed for this fringe tracking application. The use of 5 frequencies, versus some other number, depends on the level of fringe frequency distortion present in the signal, and may be tuned accordingly.
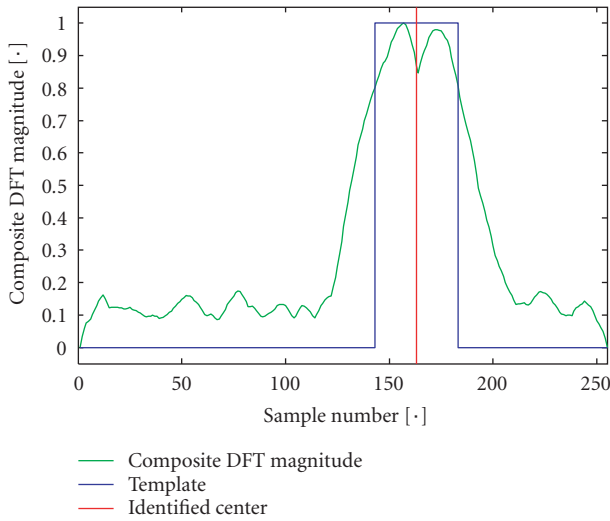
FIGURE 8: Composite DFT magnitude prior to convolution with a rectangular packet-finding template.
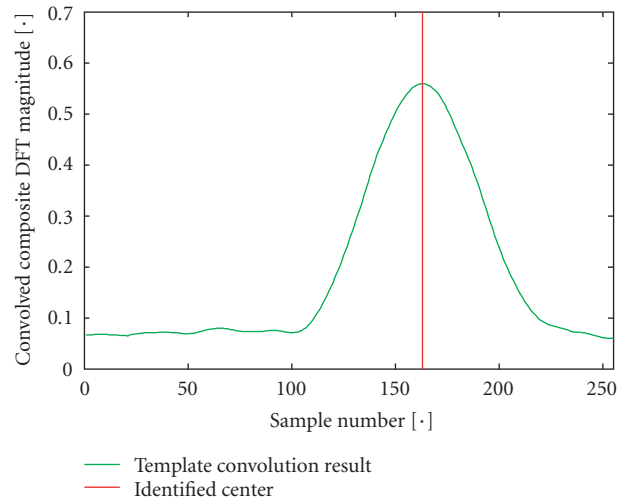

FIGURE 9: Composite DFT magnitude after convolution with a rectangular packet-finding window.

2.5. Convolution with packet-finding template

Figure 8 illustrates how a template is convolved with the composite DFT magnitude. The template, shown here at packet center, is very simple, composed of ones and zeros, leading to very efficient computation: additions and subtractions at the template transitions only, as the template is passed over the composite DFT magnitude—as opposed to an arbitrary convolution, which would require multiplications across the full template, repeated when centered at each point in the scan window.

The template is very loosely modeled after what the ideal DFT result should look like (abs(sinc)-like). The computation is efficient, since after the initial computation for the first sample, each additional sample calculation involves only one add and one subtract (no multiplies or divides)—corresponding to the two vertical edges on the template. The +1 region is chosen to correspond approximately to the width at half the composite DFT magnitude. The software implementation allows this width to be set easily, and an extension to make it adaptive should be feasible, although performance appears to be very robust to this number. For example, a value of 20 for the half-width (meaning the template spans 41 samples) was used to successfully track simulated fringes at IOTA with the scan set to both 30 and 15 microns, even though the 15-micron scan had a fringe packet twice as wide as that of the 30-micron scan.

Figure 9 shows the result of convolution with the rectangular template. The index corresponding to the maximum value of this result is used as the packet center estimate. The index corresponds to the point where the correlation between the composite DFT magnitude and the template is maximized.

2.6. Confidence metric calculation

Once the fringe packet center has been identified, a decision must be made as to the result's level of validity and degree to
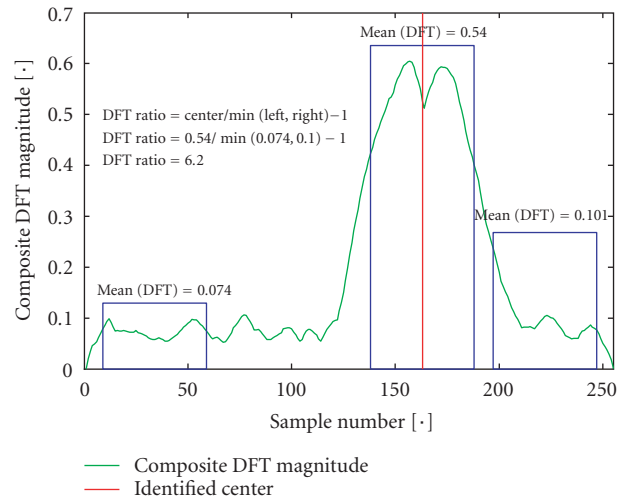

FIGURE 10: Confidence metric calculation.

which it should be used to update the scan-center position. In cases where the fringe packet disappears momentarily, it may be better to do nothing (keep scanning in the same location) than to chase the noise.

This calculation is shown graphically in Figure 10. The concept is that the DFT calculation near the identified center should have a measurably higher value than the DFT calculation on the background noise. The mean of the DFT in windows spanning 20% of the scan width is calculated at the left edge of the scan, right edge, and at the identified fringe packet center—as shown by the blue rectangles in the figure. The ratio of the mean DFT at the identified center to the smaller of the two edge measurements (minus one) is taken as the confidence metric. The reason to take both edges and then use the minimum is that this will give a valid background measurement even if the fringe packet falls at one edge of the scan.

One approach to using this confidence metric would be discrete: set a confidence threshold and either use or ignore the result based on that comparison. Setting the threshold value will depend on the level of tracking accuracy desired, the relative scan width, and other factors. It is a balance between the cost of accepting a wrong estimate and ignoring a valid one—these costs vary depending on the application. Some additional complications with this method are raised when considering multiaperture interferometry, where changing the scan center on a single delay line affects two interferograms.

2.7. Results on scan with lower SNR

While the preceding figures have illustrated the algorithm's functional steps, it is also useful to see how the algorithm performs on data with a lower SNR, as is shown in Figure 11 (this scan is also shown in Figure 3). To reduce clutter, some titles and axis labels are omitted—see the preceding figures for clarification. The DFT analysis clearly and effectively detects the fringe frequency in this noisy signal. The confidence metric calculation provides a meaningful comparison between the detected fringe packet and the background noise.

2.8. Optional operations

Nonrectangular template

If the approximate width of the sinc function is known, a template of such a shape could be used in place of the rectangular template used in Section 2.5. A simpler version that is far more computationally efficient than a sinc-shaped template, and only slightly less efficient than the rectangular template, would have 3 rectangular regions as shown in Figure 12. This was used as part of the implementation in February 2002, but was later changed (to the rectangular) since distortion due to edge effects was found to occur more frequently than expected due to the narrower scan width. However, it should be considered a viable option.

In summary, switching to such a nonrectangular template would produce a negligible increase in computation time (both are extremely efficient), provide slightly better ID accuracy for fringe packets away from the ends of the scan, but possibly significantly worse ID performance for fringe packets at or near the edge of the scan window.

Symmetry weighting

The interferogram should ideally be symmetric about its center. The algorithm originally contained a step that would calculate the symmetry and weight the DFT (or envelope) result accordingly. Unlike virtually all other signal processing steps, the symmetry calculation could not be implemented with exceptional efficiency, and ended up taking about double the processing time of all other operations combined. Problems were also encountered with edge effects; distortion would result when trying to calculate the symmetry when the fringe was partially off the scan window. The symmetry calculation was useful for gaining slightly more accuracy when the fringe was centered, but the combined issues of compute time and edge distortion led to its removal.

The sequence in Figure 13 shows how the symmetry checking step was applied. The top plot shows the output of the convolution of the packet-finding template with the composite DFT magnitude. The middle plot shows the symmetry calculation (smaller values indicate greater symmetry). This metric calculates the mean absolute value of the difference between the points on the left side of the window (sliding window of approximately the fringe packet half-width) and their reflected counterparts on the right side of the window. The bottom plot combines the two, dividing the left result by the middle. The index corresponding to the maximum value of this result is used as the packet center estimate. This sequence illustrates the sharpening possible with this symmetry calculation.

2.9. Potential algorithm improvements

There are some areas where known improvements in accuracy could be made at the cost of code complexity and development effort.

(i) *A priori expected position.* Using the expected fringe motion, and knowing whatever delay line motion has occurred, there will be an expected fringe center before the scan is analyzed. For example, if zero motion is assumed, and the control drives exactly to the previous center, then the expected fringe center would be in the middle of the scan. Some benefit should be gained from factoring this information in to the tracking algorithm, perhaps by multiplying a weighting function (perhaps exponentially decaying away from the center, with the decay rate based on the observed volatility of the motion and the confidence in the prior estimate) by the composite DFT magnitude. This would hopefully generally have little additional effect, but when faced with a choice between two weak peaks, would choose the one closer to the expected location.

(ii) *Prediction.* Add prediction to the a priori estimate above, as well as to the fringe tracker output, that is, present outputs are the identified position of the current scan and the associated confidence. What are really wanted are the expected position of the next interferogram and the confidence of that expectation. Although the improvement is expected to be slight, the linear proof of concept [4] shows measurable predictability of the fringe motion, even with a 3 Hz scan frequency (would be more predictable with faster scanning). The linear tests also indicate that the jump variance may be somewhat predictable—this could be incorporated into the exponential weighting function.

(iii) *Check jump size.* A simple check could be added to detect a large jump with low confidence—in which case, the search space could be narrowed to a smaller region about the expected center. This is a simpler version of the previous two algorithm improvements, and may not be relevant if those are implemented.

(iv) *Edge effect handling.* These are not handled very accurately in the present algorithm, since it is considered that they are less important (i.e., when scanning near
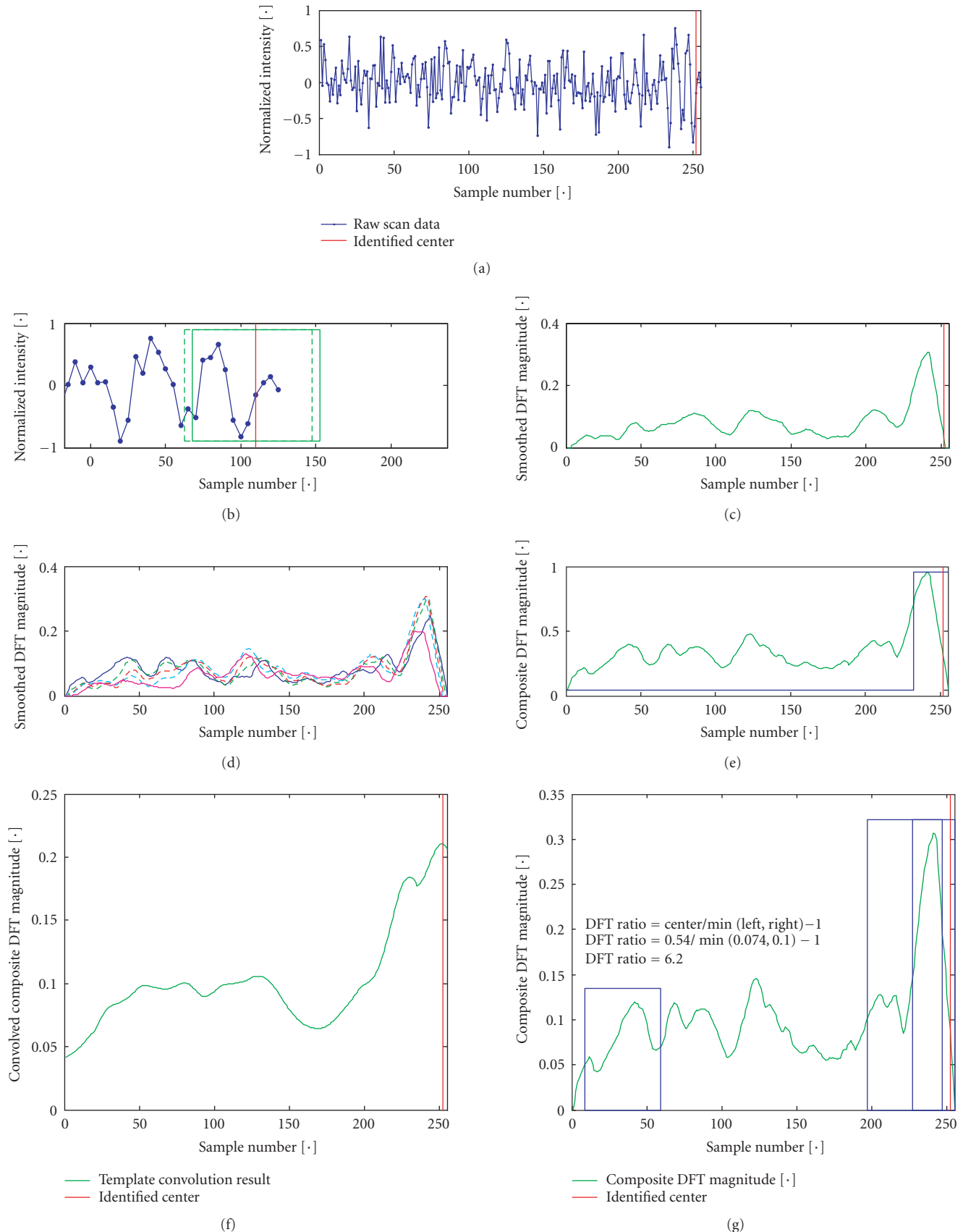
FIGURE 11: Result for scans with lower SNR ((a) scan no. 14, BC fringe, 20 April 2004; (b) DFT sliding window; (c) DFT result, 17-point window; (d) DFT results; (e) composite DFT results, template; (f) convolutions of template with composite DFT; and (g) confidence metric).
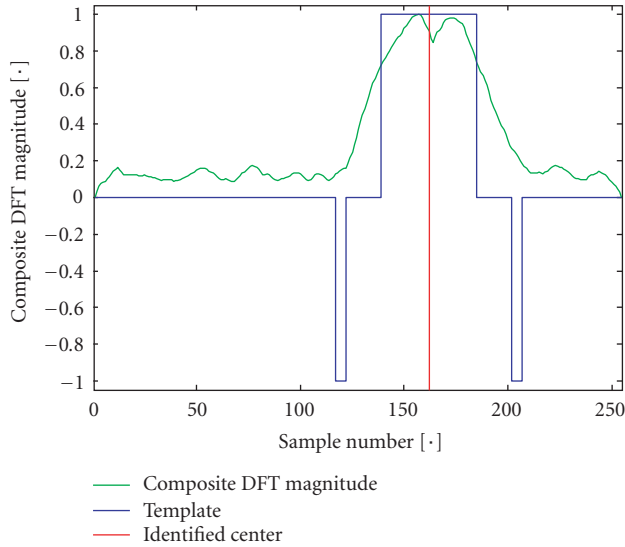
FIGURE 12: Nonrectangular template.

the edge, it is important to take a good jump towards the middle, but hitting it exactly is not important).

(v) *Adapt nominal interferogram width*. The expected width of a fringe packet is presently a fixed parameter, and the algorithm is fairly robust to adjustments to this setting. However, if adapted, the algorithm could be even more robust. If the full *ABCDE* identification is performed, the *B* (spread of sinc function) could be used to adaptively update this.

(vi) *Startup*. Since the algorithm is adaptive (the size of the DFT window is adapted from scan to scan) and the amount of adaptation permitted in a single scan is purposely limited, the performance on the first couple of scans in a new data set is sometimes not as good as it could be. If needed, allowing some extra adaptation steps on the first scan of a new data set could address this.

3. IMPLEMENTATION AT IOTA

Since the tracker needs to run on a real-time processor (Vx-Works operating system on a Motorola PowerPC 604 processor on an MVME-2431 card), after the initial development and prototyping in Matlab, the algorithm was converted (manually) to ANSI C. As the algorithm evolved during this implementation and testing process, the Matlab and C versions were continually updated to maintain the same variable names, function names, and structure to the extent possible. The two versions produce results that are identical when compared to the limit of floating point precision.

3.1. Testing

Initial testing was performed during February 2002 on the IOTA system, tracking fringes generated by a light source. Tracking performance was very good, even with temporary
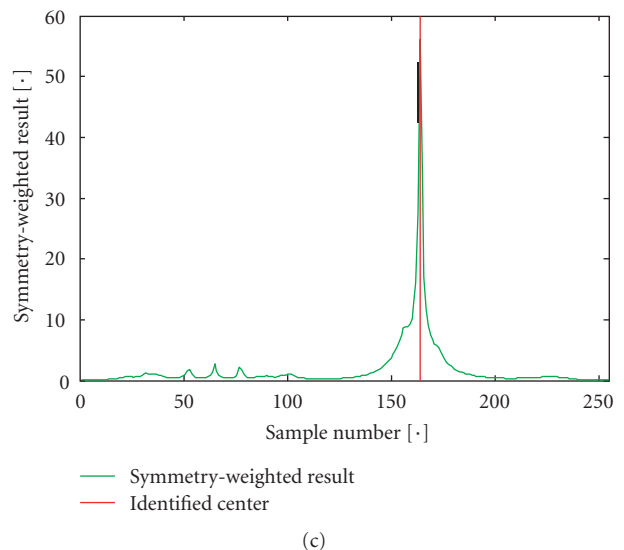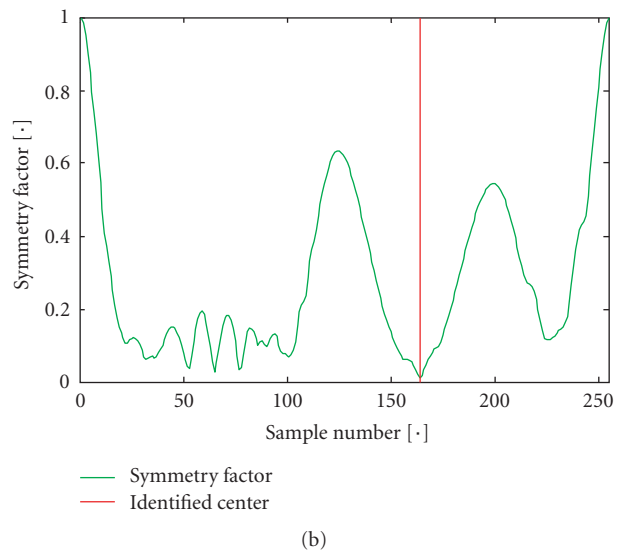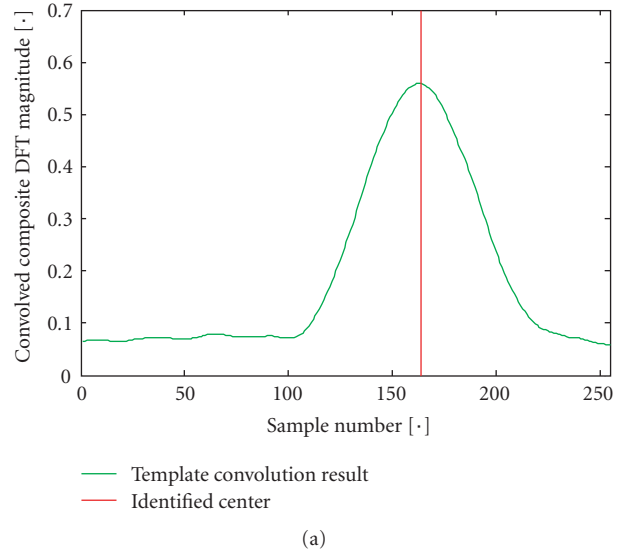

(a)

(b)

(c)

FIGURE 13: Symmetry-weighted calculation.

TABLE 1: Computation-time summary for IOTA implementation.

Time(%)	Time (ms)	Algorithm step(s)
75%	0.50	4 extra DFT calculations for window size adaptation
19%	0.13	Required DFT calculation
6%	0.04	Everything else (template, confidence, etc.)
100%	0.67	Total time per interferogram per scan

loss of fringe data (e.g., caused by banging the table)—in these cases, the system correctly decided that confidence was low and did not try to track until the fringe packet reformed. Also, the system performed very well with the scan travel set at both 15 and 30 microns, and with no manual adjustment of parameters. The fringe packet appears twice as wide for the 15-micron scan, further indicating the robustness of the algorithm. Unfortunately, due to poor weather conditions, we were unable to test it on the sky.

3.2. Speed

The computation-time results, summarized in Table 1, were obtained by running the function for 10 000 or 100 000 times and measuring the total elapsed time, both on the real-time processor and on a PC used for testing. The algorithm generated an ideal fringe packet (time to compute this is included) and then identified the center. Running the full algorithm presented in Section 2 took 0.67 millisecond per cycle on the PowerPC. Including all three interferograms for each scan, the total compute time is 2.0 milliseconds. As noted previously, there are aspects of the algorithm that could have been further optimized (e.g., pre-calculation of the sinusoidal vectors) if this compute time had been excessive. As implemented, the speed of the algorithm was sufficiently fast that compute time was negligible as compared to the scanning period [22].

3.3. Performance

The performance of the algorithm was tested extensively on offline data, although with the limitation that the true fringe packet center is not known. It was then tested briefly in an online implementation, producing good stable fringe tracking. Ettore Pedretti, in an attempt to quantitatively evaluate the online performance of this and two other algorithms [2, 23], developed an experimental procedure to do so, as described in [22]. The basic result was that this adaptive DFT-based algorithm and one developed by Pedretti both were found to perform well on both moderate and low photon flux targets, as measured by the closure-phase error. The third algorithm compared did not perform well in the low photon flux case, but worked well on moderately faint stars. Details on the testing are presented in [22].

4. CONCLUSIONS

An algorithm to perform online interferogram center identification has been developed, implemented, and tested at IOTA. It works very well on the data sets tested so far, including 1997 data, data generated by a light source using the

IOTA configuration as of February 2002, and actual on-the-sky fringes from 2002 and 2004. The adaptive nature of the DFT-based algorithm virtually eliminates the need to set any target-dependent parameters, and provides robust, accurate tracking in the presence of significant atmospheric distortion.

Initial online implementation of this algorithm at IOTA was completed in May 2002, using all three telescopes. The efficient algorithm design resulted in a compute time for all three interferograms of 2.0 milliseconds, when implemented in ANSI C on the PowerPC 266 MHz real-time processor. Fringe tracking was considered successful, and compared favorably with alternate fringe tracking approaches in a series of online experiments in May–June 2004 at IOTA.

The fringe-tracking algorithm described here may have use in fields outside astronomical interferometry, for example, in the area of thin-film ellipsometry where a white light source and Fourier transform spectrometer can be used to measure interference fringes formed by reflection from thin films and substrates.

REFERENCES

[1] R. Millan-Gabet, *Investigation of Herbig Ae/Be stars in the near-infrared with a long baseline interferometer*, Ph.D. thesis, University of Massachusetts, Amherst, Mass, USA, 1999.

[2] E. Pedretti, *Systèms d'Imagerie Interférométriques (Imaging Interferometric Systems)*, Ph.D. thesis, Université de Provence, Aix-Marseille I, Observatoire de Haute-Provence, France, 2003.

[3] E. Wilson and R. W. Mah, "Online fringe tracking and prediction at IOTA," in *Proc. 18th Congress of the International Commission for Optics*, vol. 3749 of *Proceedings of SPIE*, pp. 691–692, San Francisco, Calif, USA, August 1999.

[4] E. Wilson, E. Pedretti, J. Bregman, R. W. Mah, and W. A. Traub, "Adaptive DFT-based fringe tracking and prediction at IOTA," in *Astronomical Telescopes and Instrumentation Symposium: New Frontiers in Stellar Interferometry*, vol. 5491 of *Proceedings of SPIE*, pp. 1507–1518, Glasgow, Scotland, UK, June 2004.

[5] W. A. Traub, N. P. Carleton, J. Bregman, et al., "The third tele-scope project at the IOTA interferometer," in *Interferometry in Optical Astronomy*, vol. 4006 of *Proc. SPIE*, pp. 715–722, March 2000.

[6] W. A. Traub, A. Ahearn, N. P. Carleton, et al., "New beam-combination techniques at IOTA," in *Interferometry for Optical Astronomy II*, vol. 4838 of *Proceedings of SPIE*, pp. 45–52, Waikoloa, Hawaii, USA, August 2003.

[7] I. L. Porro, W. A. Traub, and N. P. Carleton, "Effect of tele-scope alignment on a stellar interferometer," *Applied Optics*, vol. 38, no. 28, pp. 6055–6067, 1999.

[8] N. D. Thureau, *Contribution a l'interferometrie optique a longue base en mode multi-tavelures*, Ph.D. thesis, Faculte des sciences, Universite de Nice-Sophia Antipolis, France, 2001.

[9] R. C. Jennison, "A phase sensitive interferometer technique for the measurement of the Fourier transforms of spatial bright-ness distributions of small angular extent," *Monthly Notices of the Royal Astronomical Society*, vol. 118, no. 3, pp. 276–284, 1958.

[10] J. E. Baldwin, M. G. Beckett, R. C. Boysen, et al., "The first images from an optical aperture synthesis array: mapping of Capella with COAST at two epochs," *Astronomy & Astro-physics*, vol. 306, pp. L13–L16, February 1996.

[11] G. T. van Belle, D. R. Ciardi, R. R. Thompson, R. L. Akeson, and E. A. Lada, "Altair's oblateness and rotation velocity from long-baseline interferometry," *Astrophysical Journal*, vol. 559, no. 2, part 1, pp. 1155–1164, 2001.

[12] M. Swain, G. Vasisht, R. Akeson, et al., "Interferometer ob-servations of subparsec-scale infrared emission in the nucleus of NGC 4151," *Astrophysical Journal*, vol. 596, no. 2, part 2, pp. L163–L166, 2003.

[13] A. A. Michelson and F. G. Pease, "Measurement of the diam-eter of alpha Orionis with the interferometer," *Astrophysical Journal*, vol. 53, pp. 249–259, May 1921.

[14] A. Labeyrie, "Interference fringes obtained on VEGA with two optical telescopes," *Astrophysical Journal*, vol. 196, part 2, pp. L71–L75, March 1975.

[15] F. Vakili and L. Koechlin, "Aperture synthesis in space—Computer fringe blocking," in *New Technologies for Astron-omy*, vol. 1130 of *Proceedings of SPIE*, Paris, France, April 1989, (A90-37976 16-89).

[16] Y. Rabbia, S. Menardi, J. Gay, et al., "Prototype for the Euro-pean Southern Observatory VLTI fringe sensor," in *Amplitude and Intensity Spatial Interferometry II*, J. B. Breckinridge, Ed., vol. 2200 of *Proceedings of SPIE*, pp. 204–215, Kona, Hawaii, USA, March 1994.

[17] S. Robbe, B. Sorrente, F. Cassaing, et al., "Active phase sta-bilization at the I2T: implementation of the ASSI table," in *Amplitude and Intensity Spatial Interferometry II*, J. B. Breckin-ridge, Ed., vol. 2200 of *Proceedings of SPIE*, pp. 222–230, Kona, Hawaii, USA, March 1994.

[18] P. R. Lawson, "Group-delay tracking in optical stellar interfer-ometry with the fast Fourier transform," *Journal of the Optical Society of America {A}*, vol. 12, no. 2, pp. 366–374, 1995.

[19] L. Koechlin, P. R. Lawson, D. Mourard, et al., "Dispersed fringe tracking with the multi-r0 apertures of the Grand In-terferometre a 2 Telescopes," *Applied Optics*, vol. 35, no. 16, pp. 3002–3009, 1996.

[20] D. F. Buscher, *Getting the most out of COAST*, Ph.D. thesis, Cambridge University, Cambridge, UK, 1988.

[21] S. Morel, W. A. Traub, J. Bregman, et al., "Fringe-tracking ex-periments at the IOTA interferometer," in *Interferometry in Optical Astronomy*, vol. 4006 of *Proceedings of SPIE*, pp. 506–513, Munich, Germany, March 2000.

[22] E. Pedretti, N. D. Thureau, E. Wilson, et al., "Fringe tracking at the IOTA interferometer," in *Astronomical Telescopes and Instrumentation Symposium: New Frontiers in Stellar Interfer-ometry*, vol. 5491 of *Proceedings of SPIE*, pp. 540–550, Glas-gow, Scotland, UK, June 2004.

[23] N. D. Thureau, R. C. Boysen, D. F. Buscher, et al., "Fringe envelope tracking at COAST," in *Interferometry for Optical Astronomy II*, vol. 4838 of *Proceedings of SPIE*, pp. 956–963, Waikoloa, Hawaii, USA, August 2003.

[24] M. Frigo and S. G. Johnson, "FFTW: an adaptive software architecture for the FFT," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP '98)*, vol. 3, pp. 1381–1384, Seattle, Wash, USA, May 1998.

[25] A. V. Oppenheim and R. Schafer, *Discrete-Time Signal Process-ing*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1989.

Edward Wilson is the President of Intel-lization, a consulting business applying and extending intelligent systems technologies for optimization in the aerospace and met-als industries since 1995. He attended MIT from 1983 to 1987, receiving S.B. degrees in mechanical engineering and physics and an S.M. degree in mechanical engineering. He received his Ph.D. degree in mechanical en-gineering from Stanford University in 1995, conducting his doctoral research in the Aerospace Robotics Labo-ratory, and receiving a Ph.D. minor in electrical engineering. Dr. Wilson has worked at Hughes Aircraft Company; at the US Air Force Advanced Electronics Technology Center; as a Professor on a US Navy aircraft carrier (CV-64); as Research Director at Neural Applications Corporation; and as a Visiting Scholar and Lecturer in the Stanford Aero-Astro Department, teaching a course on the modeling and analysis of dynamical systems. Areas of interest and expertise include fault detection and isolation, process optimiza-tion, identification, estimation, signal processing, and other appli-cations of advanced data analysis technologies.

Ettore Pedretti earned a Ph.D. degree in physics from the Universite de Provence in 2003. Part of his Ph.D. work was done in France on the topic of pupil densifica-tion, and part was done at the Harvard-Smithsonian Center for Astrophysics on in-strumentation and observations at the CfA's Infrared Optical Telescope Array in Ari-zona. He is currently a NASA Michelson Postdoctoral Fellow in the Astronomy De-partment, the University of Michigan, Ann Arbor. His research in-terests are in high angular resolution interferometry. He is actively involved in instrument development at the IOTA and CHARA in-terferometers.

Jesse Bregman is the Deputy Chief of the Astrophysics Branch, NASA Ames Re-search Center. He received his B.S. degree in physics from SUNY, Stony Brook, in 1971, and his Ph.D. degree in astronomy from the University of California, Santa Cruz, in 1976. He has been at NASA Ames since 1976, concentrating on infrared spec-troscopy and imaging. He has helped de-velop several infrared spectrometers and imagers that were used on ground-based and airborne telescopes. Research interests include determining the molecular content of

stars, comets, planetary nebulae, and supernovae. His main research thrust has been and continues to be understanding the nature and physics of the large interstellar molecules known as polycyclic aromatic hydrocarbons, of which benzene and naphthalene are the smallest members.

Robert W. Mah (a Ph.D., applied mechanics, Stanford University) has been the Group Lead in the Smart Systems Research Lab (SSRL), the Computational Sciences Division (code IC), NASA Ames Research Center since 1993. He has supervised development of a wide range of successful intelligent system solutions including several aerospace and medical applications.

Wesley A. Traub earned a Ph.D. degree in physics at the University of Wisconsin in 1968, and has been on the staff of the Smithsonian Astrophysical Observatory, a member of the Harvard-Smithsonian Center for Astrophysics, since that time. He has published many papers in the areas of high spectral resolution studies of the stratosphere, planetary atmospheres, and the interstellar medium. He also has a strong interest in high angular resolution interferometry from the ground, using the IOTA interferometer and the Keck nulling interferometer, with publications on stellar diameters and dust disks around stars. In addition, he has published papers on detecting and characterizing extrasolar planets with coronagraphs and interferometers. In mid-2005, he will join the staff of NASA-JPL as Project Scientist for the Terrestrial Planet Finder Coronagraph.