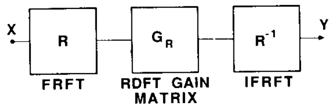

Fig. 1. Spectral filtering via FRFT.

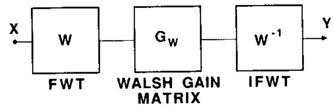

Fig. 2. Spectral filtering via FWT.

TABLE I
COMPARISON OF NUMBER OF OPERATIONS WITH RDFT AND WALSH–FOURIER METHODS OF SPECTRAL FILTERING

	Walsh–Fourier		RDFT	
N	Additions	Multiplications	Additions	Multiplications
4	18	6	15	5
8	62	22	49	15
16	198	86	141	43
32	630	342	373	115
64	2070	1366	933	291
128	7126	5462	2245	707

TABLE II
COMPARISON OF NUMBER OF OPERATIONS WITH RDFT AND WALSH–FOURIER METHODS FOR THE IMPLEMENTATION OF ZERO-PHASE FIR FILTERS

	Walsh–Fourier		RDFT	
N	Additions	Multiplications	Additions	Multiplications
4	16	4	12	4
8	52	12	40	12
16	156	44	120	36
32	460	172	328	100
64	1388	684	840	260
128	4396	2732	2056	644

siderably better for all N, with increasing margin as N grows. The reason for the growing difference can be attributed to the fact that the blocks of the Walsh gain matrix G_W, which is processed following the Walsh transform, have no simplifying structure so that they need to be directly implemented.

When the FIR filter to be implemented is zero phase, half of the elements in the diagonal blocks of G_W are zero [2]. Similarly, in the case of RDFT, $H_0(\cdot)$ is zero. Consequently, RDFT is the transform consisting of the eigenvectors of the filter matrix in this case. The number of multiplications $M_z(N)$ and the number of additions $A_z(N)$ with the RDFT method to compute RCC of size N become

$$M_z(N) = 2M_R(N) + N \tag{2.8}$$

$$A_z(N) = 2A_R(N). \tag{2.9}$$

Table II compares the number of operations with the RDFT using the above equations and the Walsh-Fourier methods [2] in the case of zero-phase FIR filters. Again, the RDFT method is considerably better for all N, and more so as N grows.

III. CONCLUSIONS

As the comparisons in Section II show, there is no good reason for using the Walsh-Fourier method in spectral filtering or for the generation of Fourier coefficients if the number of operations are the major criterion. However, the Walsh-Fourier method may still be useful if both the Walsh and the Fourier coefficients are to be utilized in a particular application.

REFERENCES

[1] C. J. Zarowski and M. Yunik, "Spectral filtering using the fast Walsh transform," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 1246–1252, Oct. 1985.
[2] A. P. Gerheim and J. W. Stoughton, "Further results in Walsh domain filtering," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, pp. 394–397, Mar. 1987.
[3] Y. Tadokoro and T. Higuchi, "Discrete Fourier transform computation via the Walsh transform," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-26, pp. 236–240, June, 1978.
[4] ——, "Conversion factors from Walsh coefficients to Fourier coefficients," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 231–232, Feb. 1983.
[5] O. K. Ersoy, "Real discrete Fourier transform," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 880–882, Aug. 1985.
[6] O. K. Ersoy and N. C. Hu, "A unified approach to the fast computation of all discrete trigonometric transforms," in *ICASSP'87 Proc.*, Dallas, TX, Apr. 1987, pp. 1843–1847.
[7] O. K. Ersoy and N. C. Hu, "Fast algorithms for the real discrete Fourier transform," in *Proc. ICASSP*, Apr. 1988, pp. 1902–1905.
[8] H. J. Nussbaumer, *Fast Fourier Transform and Convolution Algorithms*. New York: Springer-Verlag, 1982.

Exact Computation of the Unwrapped Phase of a Finite-Length Time Series

DAVID G. LONG

***Abstract*—McGowan and Kuc [1] recently showed that a direct relationship between a time series and its unwrapped phase exists. They proposed an algorithm for computing the unwrapped phase by counting the number of sign changes in a Sturm sequence generated from the real and imaginary parts of the DFT. Their algorithm is limited to relatively short sequences by numerical accuracy. An extension of their algorithm is proposed which, by using all integer arithmetic, permits exact computation of the number of multiples of π required to determine the unwrapped phase for rational-valued time sequences of arbitrary length. Since the computation is exact, the extended numerical algorithm should be of interest when accurate phase unwrapping is required.**

I. INTRODUCTION

McGowan and Kuc [1] showed that the number of multiples of π which must be added to the principal value of the phase to obtain the continuous, unwrapped phase can be uniquely determined by counting the number of sign changes in a Sturm sequence generated from a finite length sequence. Unfortunately, the numerical accuracy required for computation of the Sturm coefficients and evaluation of high-order polynomials in their algorithm precludes the application of their algorithm beyond relatively short time sequences. An extension of their algorithm is proposed which uses all integer arithmetic to permit exact numerical computation of the

Manuscript received October 27, 1986; revised March 29, 1988.
The author is with the Department of Electrical Engineering-Systems, University of Southern California, Los Angeles, CA 90089, and with the Jet Propulsion Laboratory, Pasadena, CA 91109.
IEEE Log Number 8823482.

0096-3518/88/1100-1787$01.00 © 1988 IEEE

multiples of π which must be added to the phase principal value to obtain the unwrapped phase for rational-valued time sequences of arbitrary length. In this correspondence, McGowan and Kuc's direct method is first restated in a simplified form. Then, the proposed extensions to McGowan and Kuc's algorithm are presented. A brief discussion of the tradeoffs in memory and computation required versus accuracy is presented.

II. Phase Unwrapping Using Sturm Sequences

The DFT $X(\omega)$ of the real-valued, finite-length time sequence $\{x(n), n = 0, \cdots, N-1\}$ is

$$X(\omega) = \sum_{n=0}^{N-1} x(n) e^{-jn\omega}.$$

For $0 \le \omega \le \pi$, the phase of $X(\omega)$ relative to the phase at $\omega = 0$ is

$$\arg[X(\omega)] - \arg[X(0)] = -\arctan\left\{\frac{\mathrm{Im}[X(\omega)]}{\mathrm{Re}[X(\omega)]}\right\} + L(\omega)\pi.$$

The integer-valued function $L(\omega)$ indicates the number of multiples of π which must be added to the principal value of the phase of $X(\omega)$ to produce a continuous phase function, i.e., the unwrapped phase. Assuming that $X(\omega)$ has no zeros on the unit circle, as ω increases through the zeros of Re $[X(\omega)]$, $L(\omega)$ increases or decreases depending on the sign changes of Re $[X(\omega)]$ Im $[X(\omega)]$ in order to maintain a continuous phase function. Equivalently, $L(\omega)$ at ω_1 relative to ω_2 is the number of roots of Re $[X(\omega)]$ between ω_1 and ω_2 ($\omega_1 < \omega_2$) in which Im $[X(\omega)]$ Re $[X(\omega)]$ goes from positive to negative minus the number of roots in which Im $[X(\omega)]$ Re $[X(\omega)]$ goes from negative to positive. This can be computed using a Sturm polynomial sequence generated from the time sequence.

Using the relationship between Chebyshev polynomials of different kinds, the DFT of a time sequence can be expressed in terms of Chebyshev polynomials as

$$X(\omega) = e^{-j(N-1)\omega}\left[\sum_{n=0}^{N-1} p_0(n)\, U_n(\omega) + j \sin\omega \sum_{n=0}^{N-2} p_1(n)\, U_n(\omega)\right]$$

where $U_n(\omega)$ are Chebyshev polynomials of the second kind,

$$U_n(\omega) = \frac{\sin[(n+1)\omega]}{\sin\omega} \tag{1}$$

with

$$p_0(n) = \begin{cases} x(N-1) - x(N-3)/2, & n = 0 \\ [x(N-n-1) - x(N-n-3)]/2, & 1 \le n \le N-2 \\ x(N-n-1)/2, & N-2 \le n \le N-1 \end{cases} \tag{2}$$

$$p_1(n) = x(N-n-2) \qquad 0 \le n \le N-2. \tag{3}$$

A Sturm sequence of Chebyshev polynomials $\{P_0(\omega), P_1(\omega), \cdots, P_m(\omega), m = N-1\}$ can be generated from the terms of (3) which permits computation of $L(\omega)$. Define the first two polynomials of the Sturm sequence:

$$P_0(\omega) = \sum_{n=0}^{N-1} p_0(n)\, U_n(\omega)$$

$$P_1(\omega) = \sum_{n=0}^{N-2} p_1(n)\, U_n(\omega)$$

where the remainder is generated from the "negative remainder" relationship

$$P_{k-1}(\omega) = Q_k(\omega)\, P_k(\omega) - P_{k+1}(\omega)$$

such that the order of the $(k-1)$th element of the Sturm sequence is less than the kth element. Define $Q_k(\omega)$ as

$$Q_k(\omega) = q_k U_1(\omega) + r_k U_0(\omega).$$

Then, using the recursive relationship between the Chebyshev polynomials

$$U_{n+1}(\omega) = U_n(\omega)\, U_1(\omega) - U_{n-1}(\omega) \qquad n \ge 1$$

$$U_0(\omega) = 1$$

yields

$$q_k = \frac{p_{k-1}(N-k)}{p_k(N-k-1)}$$

$$r_k = \frac{p_{k-1}(N-k-1) - q_k p_k(N-k-2)}{p_k(N-k-1)}$$

$$P_{k+1}(\omega) = \sum_{n=0}^{N-k-2} p_{k+1}(n)\, U_n(\omega)$$

$$p_{k+1}(n) = \begin{cases} -p_{k-1}(0) + r_k p_k(0) + q_k p_k(1), & n = 0 \\ -p_{k-1}(n) + r_k p_k(n) + q_k[p_k(n-1) + p_k(n+1)], & 1 \le n \le N-k-2. \end{cases}$$

The polynomial division algorithm indicated in (4)–(7) is repeated until $P_m(\omega)$, $m \le N-1$ contains only constant $U_0(\omega)$ terms.

The difference between the number of sign reversals in the Sturm sequence evaluated at ω_1 and the Sturm sequence polynomials evaluated at ω_2 ($0 \le \omega_1 < \omega_2 \le \pi$) gives the number of positive to negative changes in sign of $P_0(\omega)\, P_1(\omega)$ through the zeros of $P_0(\omega)$ minus the number of positive to negative changes in sign, i.e., $L(\omega_2) - L(\omega_1)$. While ω_2 and ω_1 can be arbitrarily chosen on $[0, \pi]$, they are typically chosen at equally spaced intervals corresponding to FFT spacing.

This approach to computing the unwrapped phase clearly indicates that the unwrapped phase is unique in the sense that once a value for the phase at $\omega = 0$ is determined, all other values follow.

III. Numerical Considerations

Direct application of McGowan and Kuc's technique can result in numerical problems for long sequences. These problems exist because more digits of significance are required to represent the coefficients of the Sturm sequence polynomials than are available in ordinary or double precision floating-point representations used in high-level languages such as Fortran or C. Inaccuracies in the computation of the Sturm sequence polynomial coefficients and their evaluation at a particular ω due to the loss of least significant digits during floating-point multiplication and division can result in the value of the evaluated polynomial having an incorrect sign. When the number of sign changes in the Sturm sequence is counted, an incorrect value for $L(\omega)$ will be computed.

The sensitivity of McGowan and Kuc's algorithm to numerical accuracy can be empirically observed as the sequence length is extended. For time sequence lengths longer than 20–40 points using ordinary floating-point representations, the computed unwrapped phase estimate is very often incorrect. Further, the algorithm does not incorporate accuracy checks so that errors in computing $L(\omega)$ are undetected even for short sequence lengths. The techniques presented below eliminate the underflow problems associated with the use of floating-point computation by using all-integer arithmetic to permit exact computations of $L(\omega)$, with accuracy checks, for arbitrary sequence lengths.

IV. Algorithm Extension

McGowan and Kuc's algorithm can be extended to permit exact computation of the coefficients of the Sturm polynomials when the time sequence takes on rational values. This is a relatively mild restriction since signals are typically digitized to integer values. A sequence of rational values can always be expressed as a sequence of integers and a multiplicative constant. The multiplicative constant does not affect the phase function and can be ignored. Thus, without loss of generality, the time sequence can be further restricted to strictly integer values.

For integer-valued time sequence, the coefficients of the Chebyshev polynomials in (2) can be expressed as integers with a multiplicative constant of $\frac{1}{2}$. However, since we are interested only in the sign of the values of the Sturm sequence, we can multiply any of the Sturm sequence polynomials by positive constants without affecting the number of sign changes. Thus, the multiplicative factor of $\frac{1}{2}$ can be discarded. $p_0(n)$ is redefined as

$$p_0(n) = \begin{cases} 2x(N-1) - x(N-3), \\ \quad n = 0 \\ x(N-n-1) - x(N-n-3), \\ \quad 1 \le n \le N-3 \\ x(N-n-2), \\ \quad N-2 \le n \le N-1. \end{cases}$$

By the same reasoning, the polynomial division algorithm in (4)–(7) can be modified to eliminate the divisions in (4) and (5) by scaling them by the positive constant $p_k^2(N-k-1)$. This modification also avoids the difficulties of division by zero in (4) and (5). Equations (4)–(7) become

$$q_k = p_{k-1}(N-k)\, p_k(N-k-1)$$

$$\begin{aligned} r_k = {} & p_{k-1}(N-k-1)\, p_k(N-k-1) \\ & - p_{k-1}(N-k)\, p_k(N-k-2) \end{aligned}$$

$$p_{k+1}(n) = \begin{cases} -p_{k-1}(0)\, p_k^2(N-k-1) + r_k p_k(0) + q_k p_k(1), \\ \quad n = 0 \\ -p_{k-1}(n)\, p_k^2(N-k-1) + r_k p_k(n) \\ \quad + q_k[p_k(n-1) + p_k(n+1)], \\ \quad 1 \le n \le N-k-2. \end{cases}$$

The resulting modified Sturm sequence consists of Chebyshev polynomials with strictly integer coefficients. If integer overflow is avoided, the integer coefficients can be computed exactly. They can be arbitrarily scaled without affecting the results as long as overflow or truncation errors are avoided. The number of bits required to represent the coefficients can be reduced, at the expense of additional CPU time, by removing common factors of the coefficients of $P_k(\omega)$. This does not affect the accuracy of the result.

We now demonstrate how to evaluate the Sturm polynomial sequence with sufficient accuracy to guarantee that the elements of the evaluated Sturm sequence have the correct sign.

The inherently large dynamic range of $U_n(\omega)$, $(-\infty, \infty)$ makes it difficult to use in a numerical algorithm. However, noting that the denominator of $U_n(\omega)$, $\sin \omega$ is independent of n [(1)] and is positive for $0 \le \omega \le \pi$, the Sturm sequence polynomials can be evaluated using

$$V_n(\omega) = \begin{cases} n+1, & \omega = 0 \\ \sin[(n+1)\omega], & \omega \ne 0. \end{cases} \tag{8}$$

rather than using $U_n(\omega)$ without affecting the number of sign changes in the Sturm sequence. The smaller range of $V_n(\omega)$ reduces the propagation of numerical errors when the polynomial coefficients are truncated (discussed below).

The use of $V_n(\omega)$ simplifies evaluation of the Sturm sequence polynomials since the circular symmetry of the sine function can be exploited. This also simplifies the problem of obtaining a sufficient number of digits of significance for $U_n(\omega)$. $V_n(\omega)$ can be obtained to the desired significance by computation of the sine to the desired accuracy or from a sine table.

Evaluation of the Sturm sequence polynomials can done with all-integer arithmetic by using a D-digit truncated integer representation of $V_n(\omega)$, i.e., by defining

$$Vi_n(\omega) = \text{nearest integer}\left\{V_n(\omega)\, 10^D\right\}.$$

The fact that $|Vi_n(\omega) - V_n(\omega)\, 10^D| \le \frac{1}{2}$ can be exploited to check the accuracy of the polynomial evaluation to ensure that a sufficient number of digits in $V_n(\omega)$ have been retained to permit accurate determination of the sign of the evaluated polynomial. Note that when ω is a multiple of $\pi/2$, $V_n(\omega)$ is an integer and the Sturm sequence polynomials can be exactly evaluated using integer arithmetic.

For $w \ne 0$ define,

$$P_k(\omega) = \sum_{n=0}^{N-k-1} p_k(n)\, V_n(\omega)\, 10^D$$

$$P_k^i(\omega) = \sum_{n=0}^{N-k-1} p_k(n)\, Vi_n(\omega)$$

$$A_k = \sum_{n=0}^{N-k-1} a_k(n)$$

$$a_k(n) = \begin{cases} |p_k(n)|, & |V_n(\omega)| \ne 1 \text{ or } 0 \\ 0, & \text{otherwise.} \end{cases}$$

$P_k(\omega)$ represents the ideal value for the evaluated Sturm polynomial element, while $P_k^i(\omega)$ is an integer-valued approximation. Note that terms for which $|V_n(\omega)| = 1$ or 0 can be computed exactly. A_k provides error bounds for $P_k(\omega)$:

$$|P_k^i(\omega)| - A_k \le |P_k(\omega)| \le |P_k^i(\omega)| + A_k.$$

When $|P_k^i(\omega)| > A_k$, $P_k^i(\omega)$ will have the correct sign. If, however, $|P_k^i(\omega)| \le A_k$, then the correctness of the sign of $P_k^i(\omega)$ cannot be guaranteed. To ensure accuracy when evaluating the Sturm sequence, A_k can be computed and checked against $|P_k^i(\omega)|$. If this check fails, D must be increased to guarantee that $P_k^i(\omega)$ has the correct sign. The minimum D to guarantee the correct sign of $P_k^i(\omega)$ depends on the value of ω and the time sequence.

The number of sign changes in the modified Sturm sequence $\{P_k^i(\omega)\}$ at $0 < \omega$ and $\omega = 0$ is used to compute $L(\omega)$. The principal value of the phase at ω can be computed using the first two terms of the modified Sturm sequence evaluated at ω. The unwrapped phase is

$$\arg[X(\omega)] - \arg[X(0)] = \arctan\left\{2\frac{P_1^i(\omega)}{P_0^i(\omega)} \sin\omega\right\} + L(\omega)\pi - (N-1)\omega.$$

V. Accuracy Versus Computation Tradeoffs

For extremely long sequences, the number of bits required to exactly represent the integer coefficients of the Sturm sequence polynomials may become large. Since only the sign of the evaluated polynomial is needed, the amount of storage and computation can be reduced, with some loss in accuracy, by truncating off the some of least significant bits of the coefficients the $P_k(\omega)$'s. The errors introduced by truncating the coefficients can lead to errors in the sign of the evaluated polynomial. However, this can be controlled by selecting the number of bits truncated. Techniques similar to the one used for checking the accuracy of the polynomial evaluation can be used to bound the resulting error and ensure numerical accuracy.

VI. Conclusion

Using integer arithmetic, the proposed modification of McGowan and Kuc's technique for phase unwrapping permits exact computation of the coefficients of the Sturm polynomial sequence for arbitrary sequence length. By careful coding, the Sturm sequence can be evaluated with sufficient accuracy to guarantee the accuracy of $L(\omega)$. The algorithm has been coded and tested using long random sequences (greater than 100 samples). In each case, $L(\omega)$ was exactly computed even when the sequence had multiple roots close to the unit circle. Since the computation is exact, the extended algorithm should be of interest when accurate phase unwrapping is required.

Reference

[1] R. McGowan and R. Kuc, "A direct relation between a signal time series and its unwrapped phase," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, pp. 719–726, Oct. 1982.

Recognition and Velocity Computation of Large Moving Objects in Images

SABRI A. MAHMOUD, MOSTAFA S. AFIFI, AND ROGER J. GREEN

Abstract—**This work presents a method for motion detection and velocity computation of large moving objects in time sequences of images. The presented algorithm and analytical formulation of the model of these large moving objects show the applicability and efficiency of this method.**

I. Introduction

Motion detection and velocity computation of moving objects in time-varying images is becoming increasingly important in diverse applications. Tracking of multitargets from video data, dust storms or cloud tracking in weather forecasts, highway traffic monitoring, and control of autonomous vehicles and robots are a few examples.

This work presents a method for motion detection and velocity computation of large moving objects of any velocity in time sequences of images. The analytical formulation for computing the velocities of large moving objects in a time sequence of images is addressed and an algorithm for the velocity computation of these moving objects is presented.

Most researchers of time-varying images use only two or three frames of a sequence. The analysis based on a few frames misses complete information about the motion of objects. References [8] and [10] show that the human visual system requires an extended frame sequence to recover the structure of moving patterns as longer sequence of frames allows the use of velocity information in solving the problem. Researchers [6], [9] have used segment and match techniques to acquire velocity information [6], [9]. This technique is sensitive to segmentation errors, and the success of the algorithm is based on accurate segmentation of static frames which is rarely satisfied in real world scenes. References [4] and [5] used differencing techniques to extract images of moving objects in a sequence. The most important feature of this technique is its simplicity and efficiency. It has limitations as it requires the images to be exactly registered, illumination to be invariant, and the moving objects should be totally displaced. References [1] and [3] used optical flow which is determined by obtaining the velocity vector for each pixel in the image. These approaches face the problem of selecting interesting points and their features as the stationary background may have many interesting points. Reference [2] used Fourier methods by applying three-dimensional Fourier transforms and used filters for velocity detection. The target is clearly detected if its size is one pixel only. Moving objects, however, are not limited to one pixel in size nor to one pixel per frame in velocity.

In this work, two one-dimensional time sequences are generated from the projections of the two-dimensional sequence on the x and y axes. Then the two-dimensional fast Fourier transform for the generated time sequences is computed. A peak in the spectrum for the selected spatial frequency is detected. The temporal frequency at which the peak is detected gives an estimate of the velocity of the moving object. Section II covers the analytical formulations for large moving objects in a time sequence with zero background, and Section III presents an algorithm for velocity computation.

Manuscript received September 19, 1987; revised December 9, 1987.
The authors are at P.O. Box 51405, Riyadh 11543, Saudi Arabia.
IEEE Log Number 8823489.

II. Analytical Formulation

Moving bodies in a sequence of images can be modeled as a big moving object. In order to simplify the description of image processing of a specific object, a binary mask $0(x, y, t)$ is introduced so that the image signals are identified as time-varying functions. The relationship between the mask and the image data is given by

$$g(x, y, t) = 0(x, y, t) f(x, y, t) + [1 - 0(x, y, t)]\, b(x, y, t) \tag{1}$$

where $g(x, y, t)$ is the recorded time sequence of images, $f(x, y, t)$ is the moving object in the time sequence, $b(x, y, t)$ is the time-varying background, $0(x, y, t) = 1$ for all pixels corresponding to the moving object, and $=0$ otherwise.

In this work, the case of zero background is considered. Using the projections of the moving object in the x and y directions, the two-dimensional sequence is transformed to two one-dimensional sequences. The velocity V in the $x - y$ plane is computed from the two components V_x and V_y such that $V = [V_x V_y]^T$:

$$g(x, t) = \sum_{y=0}^{y=M-1} g(x, y, t), \qquad g(y, t) = \sum_{x=0}^{x=N-1} g(x, y, t) \tag{2}$$

where N and M are the number of pixels of the image in the x and y directions, respectively.

In previous work [7], a method was presented for computing the velocities of two moving objects in a time sequence of images, each of one or subpixel in size, in a specific direction. This work extends the previous analysis to address objects of several pixels in size. A model for this is given by

$$g[n, m] = \sum_{i=i}^{i=r} A_i\, \delta[n - L_i - (m - m_i) V_i] \tag{3}$$

where A_i, L_i, m_i, V_i, r are the amplitudes, the initial positions, the time frames at which the objects enter the sequence, the velocities of the two moving objects, and the size of the moving object in pixels, respectively, with $\delta[\ ]$ as the Dirac impulse function.

Taking the two-dimensional discrete Fourier transform of (3) and expanding the formulation, we get

$$G[K, f] = \sum_{i=i}^{i=r} A_i M_i \frac{\text{sinc}\,[(f_i/N) + (f/M)] M_i}{\text{sinc}\,[(f_i/N) + (f/M)]} \cdot \exp(-j2\pi K L_{oi}/N) \exp\left(-j\pi[(f_i/N) + (f/M)] \cdot [M_i + 2m_i - 1]\right). \tag{4}$$

The velocity of motion as given in [7] and deducible from this equation is $V = -F_p/K_s$ where F_p is the frequency at which the peak of the spectrum is detected and K_s is the corresponding spatial frequency.

Similarly, applying the two-dimensional Fourier transform to $g(x, t)$,

$$G[K_x, f_x] = \sum_{i=i}^{i=r} A_{xi} M_{xi} \frac{\text{sinc}\,[(f_{xi}/N) + (f_x/M)] M_{xi}}{\text{sinc}\,[(f_{xi}/N) + (f_x/M)]} \cdot \exp(-j2\pi K_x L_{xoi}/N) \exp\left(-j\pi[(f_{xi}/N) + (f_x/M)][M_{xi} + 2m_{xi} - 1]\right). \tag{5}$$

0096-3518/88/1100-1790$01.00 © 1988 IEEE