Demodulation of Interferograms based on Particle Swarm Optimization

Julio Jiménez, Humberto Sossa, Francisco Cuevas, and Laura Gómez

Abstract—A parametric method to carry out fringe pattern demodulation by means of a particle swarm optimization is presented. The phase is approximated by the parametric estimation of an *n*th-grade polynomial so that no further unwrapping is required. A particle swarm is used to optimize the input parameters of the function that estimates the phase. A fitness function is established to evaluate the particles, which considers: (a) the closeness between the observed fringes and the recovered fringes, (b) the phase smoothness and c) the prior knowledge of the object, such as its shape and size. The swarm of particles evolves until a fitness average threshold is obtained. We demonstrate that the method is able to successfully demodulate fringe patterns and even a one-image closed-fringe pattern.

Index Terms—Phase retrieval; fringe analysis; optical metrology; evolutionary technique.

I. INTRODUCTION

INTERFEROMETRY is an non-destructive optical technique: It is used to measure physical variables (stress, temperature, acceleration, curvature, and so on), and this with a high degree of resolution, as it follows from the wavelength magnitude used by the light [1]. A typical interferometer splits a laser beam using a beam divisor. Beam A is called reference, and it is projected directly onto a film or a CCD camera using mirrors or optical fiber; beam B interacts with the physical phenomenon to be measured. The interaction modifies the optic path of beam B; which is then projected onto the same film or CCD camera as beam A. A diagram of the Michelson interferometer is shown in Fig. 1, and the basic operation of the interferometer is as follows. Light from a light source is split into two parts, with one part of the light travelling a different path length than the other. After traversing these different path lengths, the two parts of the light beam are brought together to interfere with each other and the interference pattern can be seen on a screen.

In optical metrology, it is well known that in a fringe pattern can be represented through total irradiance, using the following mathematical expression:

$$I(x,y) = a(x,y) + b(x,y)\cos\left[\phi(x,y)\right], \qquad (1)$$

where x, y are integer values representing coordinates of the pixel location in the fringe image, $a(x,y)$ is the background illumination, $b(x,y)$ is the amplitude modulation (e.g., this factor is related with the surface reflectance), and $\phi(x,y)$ is the phase term related to the physical quantity being measured, and is the most important term for optical metrology.

The purpose of any interferometric technique is to determine the phase term, which is related to the physical quantity being measured. Fig. 2(a) shows an interferogram, with its and its associated phase term $\phi(x,y)$ in Fig. 2(b).

One way to calculate the phase term $\phi(x,y)$ is by means of the phase-shifting technique (PST), as described in [2, 3, 4, 5].
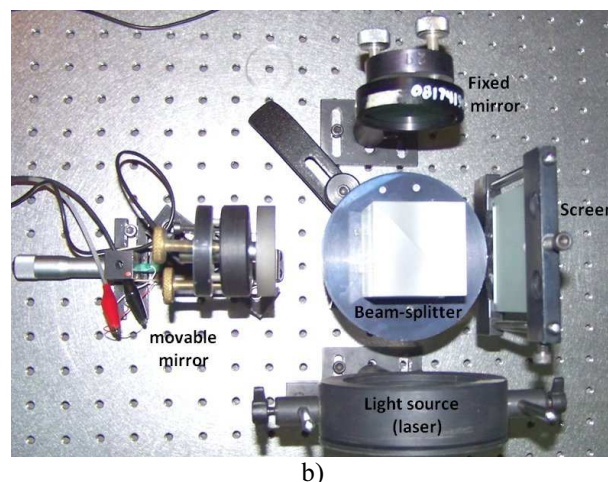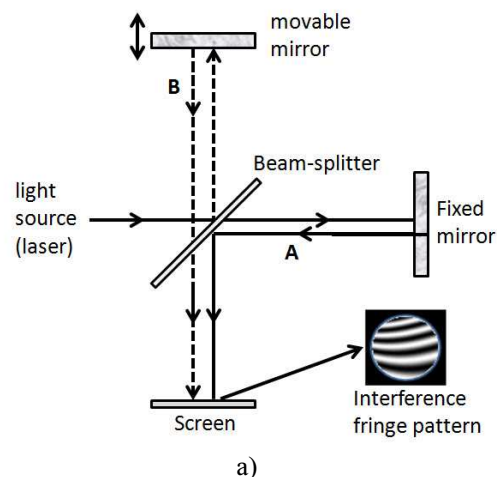

Fig. 1. a) Schematic illustration of a Michelson interferometer, and b) real schematic.

Julio Jimenez, Humerto Sossa and Laura Gomez are with Centro de Investigación en Computación, Instituto Politécnico Nacional, México DF., 07738, Mexico (email: jfvielma@cio.mx,hsossa@cic.ipn.mx, lenis45@hotmail).

Francisco Cuevas is with Centro de Investigaciones en Óptica, León, Guanajuato, 37150, Mexico (email: fjcuevas@cio.mx).

Julio Jiménez, Humberto Sossa, Francisco Cuevas, Laura Gómez

Fig. 2. (a) Fringe pattern, and (b) its phase map.

Fig. 3. Update particle.

A drawback is that it requires at least three interferograms with the phase-shifted. The phase shift among interferograms must be known and experimentally controlled. This technique can be used when mechanical conditions are met throughout the interferometric experiment, one of these conditions and the most important refer the object of study, this must be static, to project the phased shifted into it, these because they must have at least three images.

On the other hand, when the mentioned stability conditions are not satisfied, for example objects or experiments that change over time (also called transients), and for that measurement which is only possible to obtain a single image; other techniques can be used to estimate the phase term (or also known as demodulation) from a single fringe pattern; for example in [6] and [17], authors use the Fourier Transform method, while in [8], the Synchronous method is introduced, and the phase locked loop method (PLL) in [9]. However, these techniques work well only if the analyzed interferogram has a carrier frequency and a narrow bandwidth, and the signal has low noise; moreover, these methods do not perform well for phase calculation in a closed-fringe pattern. Additionally, the Fourier and synchronous methods estimate the wrapped phase due to use of an arctangent function during the phase calculation, so an additional unwrapping procedure is required [10]. The unwrapping process is difficult when the fringe pattern includes high amplitude noise, which causes differences greater than 2π radians between adjacent pixels ([11], [12] and [13]). In the PLL technique, the phase is estimated by following the phase changes of the input signal by varying the phase of a computer simulated oscillator (VCO), such that the phase error between the fringe pattern and VCO's signal vanishes.

Recent techniques make use of soft computing algorithms like neural networks and genetic algorithms (GA). In the neural network technique [14] and [15], a multi-layer neural network (MLNN) is trained by using fringe patterns, and the phase gradients associated with them, from calibrated objects. After the training, the MLNN can estimate the phase gradient when the fringe pattern is presented in the MLNN input.

The first method based on GAs to applied to the phase demodulation problem, was proposed by Cuevas et al. in [16], proposed the use of a fitness function based on a phase estimate by creating a surface through by the adjustment of the coefficients of a polynomial of order four, and in [17] was using the 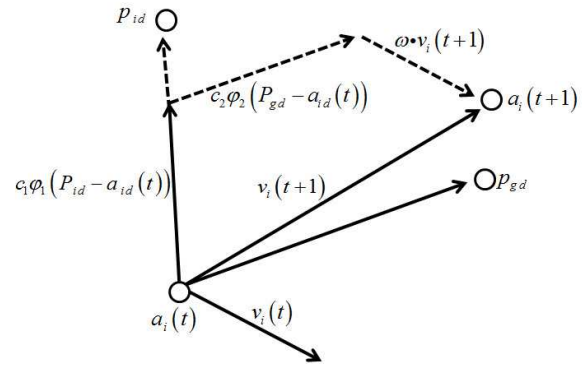fitness function created by Cuevas et al. [16], the adjustment of the surface was changed by a Zernike polynomial. To demodulate more complicated interferograms were created methods based on partition of the image, this method is called Window Fringe Pattern Demodulating (WFPD) technique, and it was proposed in [18], and used in [19], where each window is demodulated by a genetic algorithm, and these windows are slightly overlapping. The functions can be Bessel in the case of fringes coming from a vibrating plate experiment, or Zernike polynomials, in an optical testing experiment. In the case when not much information is known about the experiment, a set of low degree polynomials $p(a, x, y)$ can be used. A population of chromosomes is codified with the function parameters that estimate the phase. A fitness function is established to evaluate the chromosomes, and it considers the same aspects as the cost function in a regularization technique. The population of chromosomes evolves until a fitness average threshold is obtained. The method can demodulate noisy, closed fringe patterns and so, no further unwrapping is needed.

In this paper, we present a variation of the WFPD method introduced by Cuevas et al. in [19]. The new proposal is applied to demodulate complex fringe patterns using a particle swarm optimization technique (PSO) to fit a polynomial; it also allows one to create an automatic fringe counting based on digital image processing. In addition, we use low resolution versions of the interferogram for the recovery of the phase; in other words, we use subsampled images. Results using closed and under-sampled computer generated fringe patterns are presented.

II. PARTICLE SWARM OPTIMIZATION

Particle swarm optimization has been used to solve many optimization problems since it was proposed by Kennedy and Eberhart in [20] and [21]. After that, they published the book in [22] and several papers on this topic ([23], [24] and [25]), one of which made a study on its performance using four non-linear functions, which has been adopted as a benchmark by many researchers in this area. In PSO, each particle moves in the search space with a velocity that is in accordance with its own previous best solution and its group's previous best solution. The dimension of the search space can be any

positive integer. Following Eberhart and Kennedy's naming conventions, D is the dimension of the search space. The ith particle is represented as $A_i = (a_{i1}, a_{i2}, \ldots, a_{iD})$, and the best particle of the swarm, i.e. the particle with the lowest function value, is denoted by index g. The best previous position (i.e. the position corresponding to the best function value) of the ith particle is recorded and represented as $P_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$, and the position change (velocity) of the ith particle is $v_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$. Each particle updates its position with the following two equations:

$$v_{id}(t+1) = \omega a_{id} + c_1 \varphi_1 \left(p_{id} - a_{id}(t) \right) \\ + c_2 \varphi_2 \left(p_{gd} - a_{id}(t) \right) \qquad (2)$$

$$a_{id}(t+1) = a_{id}(t) + v_{id}(t+1), \qquad (3)$$

where for each particle i, a_i is the position, v_{id} the velocity, P_{id} the best position of a particle, P_{gd} the best position within the swarm, and c_1 and c_2 are positive constants containing the balance factors between the effect of self-knowledge and social knowledge in moving the particle towards the target; in literature, a value of 2 is usually suggested for the sum of both factors, φ_1 and φ_2 are random numbers between 0 and 1, and ω is inertia weight. Within the update of the particles, the velocity is denoted as the *momentum* with which the force is pulling the particle to continue in its current direction. The best position of a particle is the *cognitive component,* and this force emerges from the particle's tendency to return to its own best solution found so far, while the best position of a swarm is the *social component,* this is the force emerging from the attraction of the best solution found so far in its neighborhood. These features are shown in Fig. 3.

III. PSO APPLIED TO PHASE RECOVERY

As described by Eberhart and Kennedy, the PSO algorithm is an adaptive algorithm based on a social-psychological metaphor; a population of individuals (referred to as particles) adapts by returning stochastically toward previously successful regions. The fringe demodulation problem is a difficult problem to solve when the noise in the fringe pattern is high, since many solutions are possible even for a single noiseless fringe pattern. Besides, the complexity of the problem is increased when a carrier frequency does not exist (closed fringes are present).

Given that for a closed fringe interferogram there are multiple phase functions for the same pattern, the problem is stated as an *ill-posed* problem in the Hadamard sense, since a unique solution cannot be obtained [26]. It is clear that the image of a fringe pattern $I(x, y)$ will not change if $\phi(x, y)$ in (1) is replaced with another phase function $\hat{\phi}(x, y)$ given by:

$$\hat{\phi}(x, y) = \begin{cases} -\phi(x, y) + 2\pi k & (x, y) \in R \\ \phi(x, y) & (x, y) \notin R \end{cases}, \qquad (4)$$

where R is an arbitrary region, and k is an integer. In this work, PSO is presented to carry out the optimization process, where a parametric estimation of a non-linear function is proposed to fit the phase of a fringe pattern. Then, the PSO technique fits a global non-linear function instead of a local plane to each pixel, just as it is done in regularization techniques [27] and [28]. The fitting function is chosen depending on prior knowledge of the demodulation problem, such as object shape, carrier frequency, pupil size, etc; when no prior information about the shape of $\phi(x, y)$ is known, a polynomial fitting is recommended. In this paper, the authors have used a polynomial fitting to show how the method works.

The purpose in any application of PSO is to evolve a particle swarm of size P (which codifies P possible solutions to the problem) using the update velocity and position of each particle, with the goal of optimizing a fitness function that solves the problem.

In phase demodulation from fringe patterns, the phase data can be approximated by choosing from one of several fitting functions. The fitness function is modeled by the following considerations: a) the similarity between the original fringe image and the genetic generated fringe image, and b) the smoothness in the first and second derivatives of the fitting function.

A. Fitness function

The fitness function U that was utilized in this paper to evaluate the pth particle a^p in the swarm, used an *r-degree* approximation, and is given by:

$$p_r(a, x, y) = a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 y^2 + a_5 xy \\ + a_6 x^2 y + a_7 xy^2 + \cdots + a_{\left(\frac{(r+1)(r+2)}{2}\right)} y^r \qquad (5)$$

Many ways to quantify the quality of fitness function U can be used. We decided to use a term that compares the RMS error between the original fringe pattern and the fringe pattern obtained from the estimated phase:

$$U(a^p) = \sum_{y=1}^{R-1} \sum_{x=1}^{C-1} \left[I_N(x, y) - \cos\left(f(a^p, x, y) \right) \right]^2 \qquad (6)$$

where *x, y* are integer values representing indexes of the pixel location in the fringe image. Super-index p is an integer index value between 1 and p, which indicates the number of particles in the swarm. $I_N(x, y)$ is the normalized version of the detected irradiance at point (x, y).

The data from the inferferogram were normalized in the range [-1,1], $R \times C$ is the image resolution whose fringe intensity values are known, and $f(a^p x, y) = p(a^p x, y)$. Additional terms are added to the fitness functions; in this

case, the restrictions for the phase. The fitness function used by Cuevas et al. in [19] incorporates three criteria: similarity, smoothness and overlapped phase similarity with a previously estimated phase. Similarity between fringe patterns is given by equation (6), while smoothness and overlapped phase similarity are expressed by the following equation:

$$R\left(a^p\right) = \sum_{y=1}^{R-1}\sum_{x=1}^{C-1}\left\{ \lambda\left[\left(f\left(a^p,x,y\right)-f\left(a^p,x-1,y\right)\right)^2 \right. \right. \\ \left. \left. +\left(f\left(a^p x,y\right)-f\left(a^p x,y-1\right)\right)^2 \right] m\left(x,y\right) \right\} \quad (7)$$

where $R\left(a^p\right)$ is the total amount of restrictions added to the fitness function for a given window whose origin is (r,c); $m(x,y)$ is a mask that indicates where the fringe pattern appears inside the image, and λ is a smoothness weight factor (it should be clear for the reader that a higher value of parameter λ implies a smoother function to be fitted).

The third criterion is eliminated in order to simplify the fitness function to get a robust retrieval in just one window. This way, the phase in different windows can be demodulated in parallel. The phase segments are sequentially overlapped. Noise filtering and fringe normalization are solved by using alternative low-pass filtering techniques. We assume smooth phase continuity distributed in first and second derivatives.

The new fitness function can thus be written as:

$$U\left(a^p\right) = \alpha - \sum_{y=1}^{R-1}\sum_{x=1}^{C-1}\left\{ \left(I_N\left(x,y\right)-\cos\left(f\left(a^p,x,y\right)\right)\right)^2 \right. \\ + \lambda\left[\left(f\left(a^p,x,y\right)-f\left(a^p,x-1,y\right)\right)^2 \right. \\ \left. \left. +\left(f\left(a^p,x,y\right)-f\left(a^p,x,y-1\right)\right)^2 \right] \right\} m\left(x,y\right) \quad (8)$$

Parameter α must be set to the maximum value of the second term in equation (8). This is done with the aim of converting the problem from a minimal to a maximal optimization question, since a fitness function for PSO is considered to be a non-negative image of merit and profit; this is:

$$\alpha = \max_p\left(\sum_{y=1}^{R-1}\sum_{x=1}^{C-1}\left\{ \left(I_N\left(x,y\right)-\cos\left(f\left(a^p,x,y\right)\right)\right)^2 \right. \right. \\ + \lambda\left[\left(f\left(a^p,x,y\right)-f\left(a^p,x-1,y\right)\right)^2 \right. \\ \left. \left. \left. +\left(f\left(a^p,x,y\right)-f\left(a^p,x,y-1\right)\right)^2 \right] \right\} \right) m\left(x,y\right) \quad (9)$$

The first term inside the double summation in equation (9) attempts to keep the local fringe model close to the observed irradiances in the least-squares sense, while the second term is a local discrete difference, which enforces the assumption of smoothness and continuity of the detected phase.

B. Decoding particles

As it was said earlier, PSO is used to find the function parameters; in this case, vector a. If we use this function, the particle can be represented as:

$$a = \left[a_0 \; a_1 \; \ldots \; a_q \right] \quad (10)$$

A *k-bit* bit-string is used to codify a particle value; then, the particle has $q \times k$ bits in length. We define the search space for these parameters. The bit-string codifies a range within the limits of each parameter. The decoded value of the a_i parameter will use the methodology introduced by Toledo and Cuevas in [18], and is:

$$a_i = L_i^B + \frac{L_i^U - L_i^B}{2^k - 1} N_i \quad (11)$$

where a_i is the ith parameter real value, L_i^B is the ith bottom limit, L_i^U is the ith upper limit, and N_i is the decimal basis value. These maximum values can be expressed as:

$$L_0^B = -\pi, \quad L_0^U = \pi \quad (12)$$

$$L_i^U = -L_i^B \quad (13)$$

$$L_i^U = \frac{4\pi F}{R1_i^m C1_i^n} \quad (14)$$

where F is twice the maximum number of fringes on the window; the equation is expressed in [18]:

$$F = 2 \times \max\left(F_x, F_y, \sqrt{F_x^2 + F_y^2} \right) \quad (15)$$

F_x and F_y are the maximum fringe numbers in the x and y directions. Finding the value for F automatically is not an easy problem to solve; to our knowledge, there are several algorithms that perform this count, ranging from manual counting by an expert, using a priori knowledge of the phenomenon being measured, even those based on image processing, so in this paper, to get the maximum number of fringes in an image, we propose combining image thresholding described in [29] and connected component labeling, as described in [31] and [32]:

Image thresholding: To binarize the fringe image, we have used Otsu's technique [29], which is known to be based on discriminated analysis. The threshold value t obtained by this method allows partitioning the image into two classes: C_0 and C_1 (i.e., the foreground and background). In other words: $C_0 = \{0,1,2,\ldots,t\}$ and $C_0 = \{t+1,t+2,\ldots,L-1\}$, where L is the number of gray levels. For an example of the application of Otsu's procedure onto an image, refer to Fig. 5. Fig. 6(a) shows a simple fringe image, while Fig. 6(b) shows the corresponding binary version obtained by Otsu's method.

Connected component labeling: Segmenting a binary image by means of connected component labeling is a standard procedure found in literature. A connected component (CC) is a region of foreground pixels for which a connected path can be found for any two pixels belonging to the region. Finding the connected components in a binary image can be done in many ways ([30], [31] and [32]). The simplest method consists in iteratively replacing each label with the minimum of its 8-connected neighborhood [31]. The algorithm begins with an initial labelling of all 1-pixels, and ends when no more replacements can be made.

In this work, we use the following methodology. Taking as input the binary image, for example the image shown in Fig. 5(a), the algorithm makes a journey through the image from left to right and top to bottom. At each position, a 2×2 neighborhood is analyzed. The positions of the pixels in the neighborhood are: $a_{(i,j)}$, $a_{(i+1,j)}$, $a_{(i,j+1)}$ and $a_{(i+1,j+1)}$ (see Fig. 4). Under 8-connectivity, it is guaranteed that the four pixels are connected.
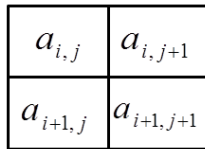
$a_{i,j}$	$a_{i,j+1}$
$a_{i+1,j}$	$a_{i+1,j+1}$

Fig. 4. 2x2 neighborhood scheme.

To assign the subset that corresponds to each pixel, the following steps are applied:

1. Check the validity of $a_{(i,j)}$, $a_{(i+1,j)}$, $a_{(i,j+1)}$ and $a_{(i+1,j+1)}$. A pixel is valid if $I(i,j)=1$ (it belongs to the foreground), or zero if $I(i,j)=0$ (it belongs to the background).
2. Of the pixels that are valid, check whether one of them has been previously assigned to a given neighborhood. If one or more of the valid pixels have been assigned to a neighborhood, then search for the pixel with the highest number of elements. This is done by using a vector T, which contains all the subsets that have already been assigned, as well as the number of elements in each subset. This facilitates the search.
3. Among the pixels of the neighbourhood that are valid, we search for the pixel whose subset has more elements. To this subset, the other pixels will be assigned.
4. If none of the pixels is assigned to a group, then assign them to a new subset and update the value of the tag in the vector T.
5. Update the values of the subsets and advance one pixel to repeat the steps above.
6. Repeat these steps all over the image.

The result of applying this methodology to an image is shown in Fig. 6(c). The four connected foreground regions

appear in different colors. The number of connected components found is a good approximation of the maximum number of fringe patterns in the image.

Alternative way to compute the maximum number of fringes in an image: Another way to find the maximum number of fringes F is as follows. Starting from the central pixel of the fringe image, scan it horizontally, vertically and diagonally, in both directions, as shown in Fig. (7). As an example, in this figure, when we go from the central point to the right, we find a transition from the fringe to the background; as we continue we find a second transition from the background to another fringe. We have thus two fringes. To get the final number, we take into account the considerations given in [1]; by using the interference order for each fringe, we arrive at the end of the swapping that in the example image there are four fringes. This value F can now be used in equation (14) to compute L_i^U. From Equation (13), we can compute L_i^B. Finally, we can substitute these two values in Equation (11) to estimate each a_i. This constitutes an original and very simple procedure to find the components of vector a.
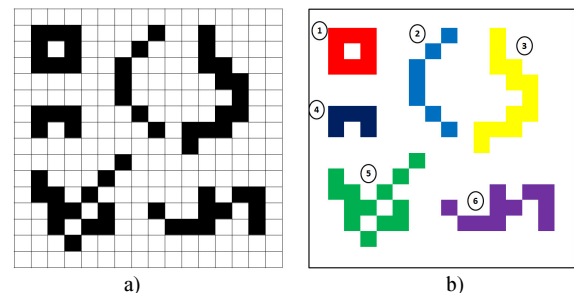

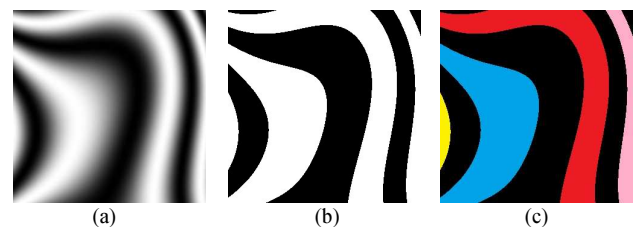Fig. 5. Example of connected component labelling, a) original image, b) labelling image.

Fig. 6. (a) image of fringe patterns, (b) binary image using Otsu method, (c) labelling image with the result of 4 fringes in the image.
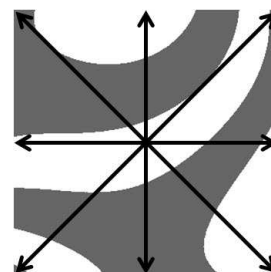

Fig. 7. Direction of sweeps for the location of fringes.

TABLE I
TABLE OF INERTIA AND VELOCITY PARAMETERS

Inertia	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.0001	2.870	3.432	3.612	3.505	3.839	3.277	2.916	2.777	2.395
0.0002	3.007	3.044	3.210	3.083	2.725	2.680	1.688	1.801	2.366
0.0003	1.665	1.875	2.565	2.559	1.576	1.708	**1.151**	1.945	2.469
0.0004	2.170	**1.738**	2.777	1.912	**1.290**	2.171	1.806	**0.567**	1.946
0.0005	1.883	1.860	2.838	1.686	1.701	2.063	1.969	0.791	1.792
0.0006	2.106	2.134	2.900	**1.086**	2.318	1.705	1.645	1.399	2.343
0.0007	1.928	1.993	**0.853**	1.168	2.019	2.270	1.772	1.428	1.828
0.0008	**0.893**	1.938	1.350	1.531	2.019	2.632	1.373	1.373	2.260
0.0009	1.536	1.911	1.436	1.773	2.407	**0.313**	1.902	0.779	**1.523**

For the special case a_0 (i=0), the limits are between $-\pi$ and $+\pi$. a_0 is eliminated from parameter vector a to redefine a new vector a':

$$a' = \begin{bmatrix} a_1 & a_2 & \dots & a_q \end{bmatrix} \qquad (16)$$

so $p(a, x, y)$ can be expressed as follows:

$$p(a, x, y) = p(a', x, y) + a_0 \qquad (17)$$

and replacing (17) into (1):

$$I(x, y) = a(x, y) + b(x, y) \cos\left[p(a', x, y) + a_0 \right], \qquad (18)$$

Additionally, a_0 can be expressed as $a_0 = 2\pi l + a'_0$, with l being an integer, and $a'_0 < 2\pi$, so equation (18) becomes:

$$I(x, y) = a(x, y) + b(x, y) \cos\left[p(a', x, y) + a'_0 + 2\pi l \right], \qquad (19)$$

The cosine function is periodical with period 2π, so:

$$I(x, y) = a(x, y) + b(x, y) \cos\left[p(a', x, y) + a'_0 \right], \qquad (20)$$

In equation (20) demonstrates that limits for a_0 within a range of 2π are enough to represent the phase of the fringe pattern.

C. Convergence

PSO convergence depends mainly on swarm size. Large swarm convergence takes place in smaller number of, but processing time is increased. To stop the PSO process, different convergence measures can be employed. In this paper, we have used a relative error comparison between the fitness function value of the best vectors in the swarm and value a as follows in equation (21), which is the maximum possible value that we can get from equation (8). Thus, we can establish a relative evaluation with uncertainty to stop PSO as:

TABLE II
BEST PARTICLES

Inertia Velocity	0.1 0.0008	0.2 0.0004	0.3 0.0007	0.4 0.0006	0.5 0.0004	0.6 0.0009	0.7 0.0003	0.8 0.0004	0.9 0.0009
Low-resolution interferogram									
High resolution interferogram									

TABLE III
WORST PARTICLES

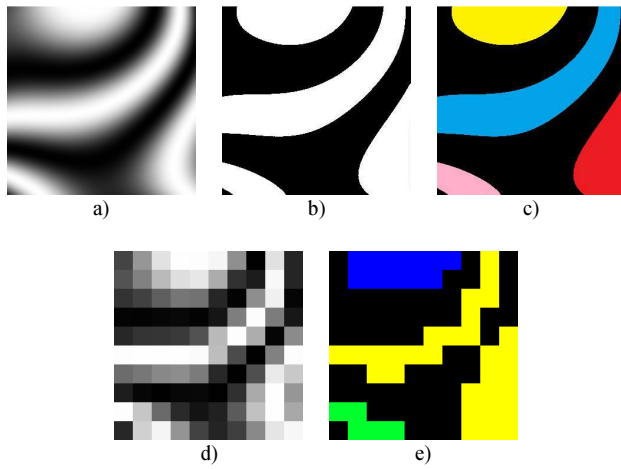| Inertia Velocity	0.1 0.0002	0.2 0.0001	0.3 0.0001	0.4 0.0001	0.5 0.0001	0.6 0.0001	0.7 0.0001	0.8 0.0001	0.9 0.0003
Low-resolution interferogram									
High resolution interferogram									

a)   b)   c)

d)   e)

Fig. 8. (a) Image of fringe pattern in resolution , b) binary image using Otsu's method, (c) labeling image with the result of 4 fringes in the image. (d) Low resolution image with sub-Nyquist,(e) labeling image with the result of 3 fringes in the image.
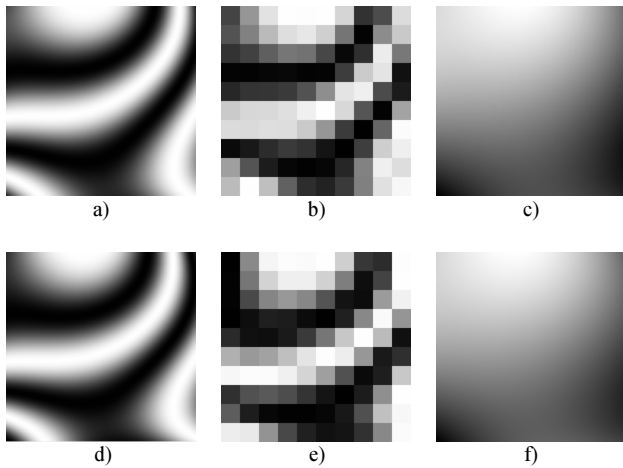

a)   b)   c)

d)   e)   f)

Fig. 9. (a) Observed fringe pattern, (b) Observed fringe pattern in low resolution, (c) its phase map. (d) estimated fringe pattern by PSO, (f) in low resolution and (g) its phase map.

$$\left| \frac{\alpha - U\left(a^*\right)}{\alpha} \right| < \varepsilon \qquad (21)$$

where $U\left(a^*\right)$ is the fitness function value of the best vectors in the swarm in the current iteration, and ε is the relative error tolerance. Additionally, we can stop the process in a specified number of iterations if equation (21) is not satisfied.

IV. EXPERIMENTS

The proposed method was applied to estimate the phase for a closed fringe pattern. We used a particle swarm size of 100, with 70 iterations, inertia was chosen in the range [0.1 to 0.9], and velocity was a number in the range [0.0001 to 0.0009]. In each particle, the coded coefficients of a fourth degree polynomial were included. The following polynomial was coded in each particle:
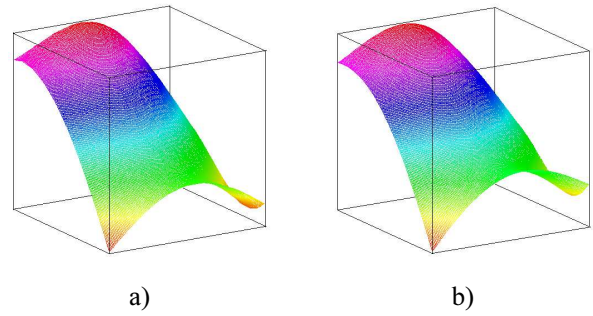

a)   b)

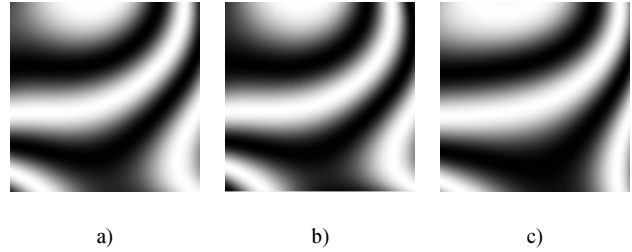Fig. 10. Phase map observed (a), and phase map estimated by PSO (b).

a)   b)   c)

Fig. 11. (a) Observed fringe pattern, (b) estimated fringe pattern by PSO, (c) estimated fringe pattern by GA.

$$\begin{aligned} p_4\left(x, y\right) = {} & a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 xy \\ & + a_5 y^2 + a_6 x^3 + a_7 x^2 y + a_8 xy^2 \\ & + a_9 y^3 + a_{10} x^4 + a_{11} x^3 y + a_{12} x^2 y^2 \\ & + a_{13} xy^3 + a_{14} y^4 \end{aligned} \qquad (22)$$

The 15 coefficients were configured in each particle inside the swarm to be evolved. As real interferograms present low contrast, and to show that our proposal performs efficiently, a low noise closed fringe pattern was generated using the following expression:

$$I\left(x, y\right) = 127 + 63 \cos\left(P_4\left(x, y\right) + \eta\left(x, y\right)\right), \qquad (23)$$

where

$$\begin{aligned} p_4\left(x, y\right) = {} & 0 - 0.7316x - 0.2801y + 0.0065x^2 \\ & - 0.00036xy - 0.0372y^2 + 0.00212x^3 \\ & + 0.000272x^2 y + 0.001xy^2 - 0.002y^3 \\ & + 0.000012x^4 + 0.00015x^3 y + 0.00023x^2 y^2 \\ & + 0.00011xy^3 + 0.000086y^4 \end{aligned} \qquad (24)$$

and $\eta\left(x, y\right)$ is the uniform additive noise in the range [–2 radians to 2 radians]. Additionally, the fringe pattern was generated with a low resolution of 10×10 pixels. In this case, we used a parameter search range of [-1 to 1]. The swarm of particles evolved until the number of iterations reached 70, and relative error tolerance ε was 0.05 in equation (21). The fringe pattern and the binary image field of the computer generated interferogram are shown in Figs. 8(a) and 8(b), respectively, with a resolution to 512×512, and after applying
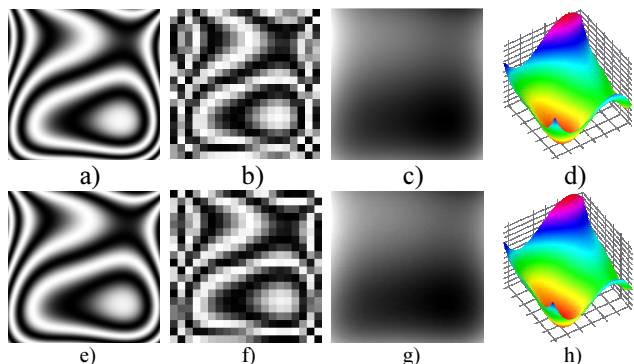
Fig. 12. (a) Observed fringe pattern, (b) observed fringe pattern in low resolution, (c) its phase map, (d) phase in 3D, e) PSO estimated fringe, (f) in low resolution and (g) its phase map and h) phase in 3D.
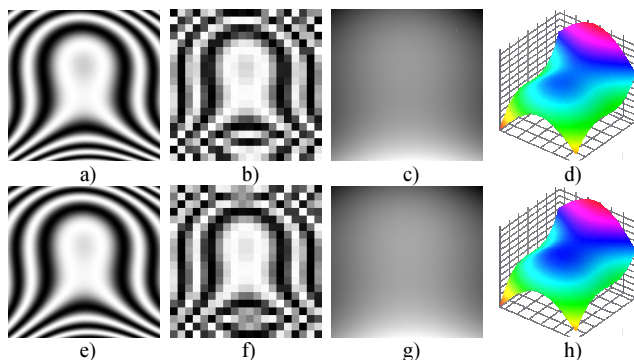

Fig. 13. (a) Observed fringe pattern, (b) observed fringe pattern in low resolution, (c) its phase, (d) phase 3d, e) estimated fringe pattern by PSO, (f) in low resolution and (g) its phase map and h) phase 3d.
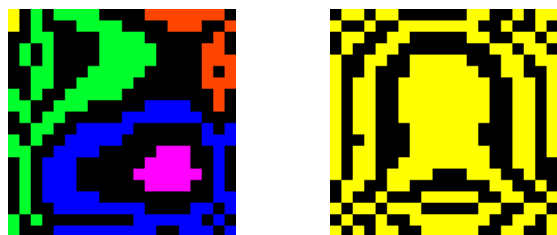

Fig. 14. Labelling images in low-resolution with the result of: (a) 5 fringes in the image show in 12(b), and 1 fringe in the image show in 13(b).

the Otsu's method, we obtain the number of fringes on the image which was 4 number of fringes, this shown in Fig. 8(c). Finally in Figs. 8(d) and Fig. 8(e) shows a sub-sampled image 10x10 and in the connected component labeled obtain the 3, number of fringes.

The fringe pattern and the phase field of the computer generated interferogram are shown in Figs. 9(a) and 9(b), respectively. The PSO technique was used to recover the phase from the fringe pattern. The fringe pattern and the phase estimated through PSO are shown in Figures 9(d), 9(e) and 9(f).

The 3D phase map observed is shown in Fig. 10(a), and the 3D phase map estimated by PSO in Fig. 10(b). Tests are

shown in Table 1, the best particles for the testers are shown in Table 2, and Table 3 shows the worst particle for the testers.

Additionally, our method was compared with that proposed by Toledo and Cuevas in [18], which is based on genetic algorithms, and in which, taking into consideration the settings of GA parameters, eight parameters were initialized: number of generations, number of population, cross and mutation rate, type of selection, mutation rate and type of cross. In our case, only four parameters were initialized: iterations (generations), swarm (population), inertia and speed. Finally, during the test an error of 0.4281 was obtained with the GA-based method. With our PSO based proposal, we obtained an error of 0.313.

The Fig. 11(a) shows the original interferogram; figures 11(b) and 11(c) illustrate the result obtained through our method and the result obtained with the GA method introduced in [19]. The interferogram demodulation, in comparison, was almost identical, but the difference is that the image input used with the PSO technique with PSO was recovered from a low level image that had a serious problem of sub-Nyquist in that it no longer distinguished fringes.

The proposed methodology was applied to other images to show its performance. For this, refer to Figs. 12 and 13.

The use of a sub-sample with a high sub-Nyquist problem is something where traditional techniques (Fourier method, Synchronous method and the phase locked loop method) fail; instead, techniques that use GAs have a sub-sampling Nyquist above the limit (one fringe per pixel), as shown in Fig. 14.

Compared with other methods in literature, our method has the advantage that, using a single image, it does not apply any unwrapping module to the phase, and that the polynomial is directly the phase term; it can work with images with high sub-Nyquist, a problem that traditional methods have so far failed to solve.

Execution time is considered fast compared to methods using GAs which is due to the encoding and the image size.

V. CONCLUSION

A PSO based technique was applied to recover the modulating phase from closed and noisy fringe patterns. A fitness function, which considers prior knowledge about the object being tested, is established to approximate the phase data. In this work, a fourth degree polynomial was used to fit the phase.

A swarm of particles was generated to carry out the optimization process. Each particle was formed by a codified string of polynomial coefficients. Then, the swarm of particles evolved using velocity, position and inertia.

The proposal works successfully where other techniques fail (Synchronous and Fourier methods). This is the case when a noisy, wide bandwidth and/or closed fringe pattern is demodulated. Regularization techniques can be used in these cases, but this proposal has the advantage that the cost function does not depend upon the existence of derivatives and restrictive requirements of continuity (gradient descent methods). Since PSO works with a swarm of possible solutions instead of with a single solution, it avoids falling into a local

optimum. Additionally, no filters and no thresholding operators were required, in contrast with the fringe-follower regularized phase tracker technique.

PSO has the advantage that if the user has prior knowledge of the object shape, then a better suited fitting parametric function can be used instead of a general polynomial function. Additionally, due to the fact that the PSO technique gets the parameters of the fitting function, it can be used to interpolate sub-pixel values and to increase the original phase resolution or interpolate where fringes do not exist or are not valid. A drawback is the selection of the optimal initial PSO parameters (such as swarm size, inertia and velocity) that can increase convergence speed.

REFERENCES

[1] D. Malacara, *Optical Shop Testing*, New York: Wiley, 1992.
[2] D. Malacara, M. Servin and Z. Malacara, *Interferogram Analysis for Optical Testing*, Marcel Dekker, Ed. New York: CRC Press, 1998.
[3] D.W. Robinson and G.T. Reid, *Interferogram Analysis: Digital Fringe Measurement Techniques*, London: IOP publishing, 1993.
[4] K. Creath, "Phase measurement interferometry techniques," in *Progress in Optics*, vol. 26, E. Wolf, Ed. Amsterdam: Elsevier, pp. 348-393, 1988.
[5] K. Creath, *Interferogram Analysis*, D. Robinson and G.T. Reid (Eds.) London: IOP Publishing, pp. 94, 1993.
[6] M. Takeda, H. Ina and S. Kobayashi, "Fourier-transform method of fringe-pattern analysis for computer based topography and interferometry," *J. Opt. Soc. Am.*, vol.72, pp. 156-160, 1981.
[7] X. Su, W. Chen, "Fourier transform profilometry: a review," *Opt. Laser Eng.*, vol. 35, pp. 263-284, 2001.
[8] K.H. Womack, "Interferometric phase measurement using spatial synchronous detection," *Opt. Eng.*, vol. 23, pp. 391-395, 1984.
[9] M. Servin and R. Rodriguez-Vera, "Two dimensional phaselocked loop demodulation of interferogram," *J. Mod. Opt.,* vol. 40, pp. 2087-2094, 1993.
[10] D.C. Ghiglia and M.D. Pritt, *Two-dimensional Phase Unwrapping*, New York: John Wiley & Sons, Inc., 1998.
[11] X. Su and L. Xue, "Phase unwrapping algorithm based on fringe frequency analysis in Fourier-transform profilometry," *Opt. Eng.*, vol. 40, pp. 637-643, 2001.
[12] D.C. Ghiglia, G.A. Mastin and L.A. Romero, "Cellular automata method for phase unwrapping," *J. Opt. Soc. Am.*, vol. 4, pp. 267-280, 1987.
[13] M. Servin, F.J. Cuevas, D. Malacara, J.L. Marroquin and R. Rodriguez-Vera, "Phase unwrapping through demodulation using the RPT technique," *Applied Optics*, vol. 38, pp. 1934-1940, 1999.
[14] F.J. Cuevas, M. Servin, O.N. Stavroudis and R. Rodríguez-Vera, "Multilayer neural network applied to phase and depth recovery from fringe patterns," *Optics Communications*, vol. 181, pp. 239–259, 2000.
[15] F.J. Cuevas, M. Servin and R. Rodríguez-Vera, "Depth recovery using radial basis functions," *Opt Commun.*, vol. 163, pp. 270–277, 1999.
[16] F.J. Cuevas, J.H. Sossa-Azuela and M. Servin, "A parametric method applied to phase recovery from a fringe pattern based on a genetic algorithm," *Opt Commun.*, vol. 203, no. 3-6, pp. 231–239, 2002.
[17] L.E. Mancilla, J.M. Carpio and F.J. Cuevas, "Demodulation of Interferograms of Closed Fringes by Zernike Polynomials using a technique of Soft Computing," *Engineering Letters,* vol. 15, no. 1, pp. 99-104, 2007.
[18] L.E. Toledo and F.J. Cuevas "Optical Metrology by Fringe Processing on Independent Windows Using a Genetic Algorithm," *Experimental mechanics*, vol. 48, pp. 559-569, 2008.
[19] F.J. Cuevas, F. Mendoza, M. Servin and J.H. Sossa-Azuela, "Window fringe pattern demodulation by multi-functional fitting using a genetic algorithm," *Opt. Commun.*, vol. 261, pp. 231–239, 2006.
[20] J. Kennedy and R.C. Eberhart, "Particle swarm optimization," in *Proc. IEEE Intl. Conf. on Neural Networks*, vol. 4. Piscataway, NJ: IEEE Service Center, pp. 1942-1948, 1995.
[21] R.C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory controllers," in *Proc. Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan, Piscataway, NJ: IEEE Service Center, pp. 39-43, 1995.
[22] R.C. Eberhart and Y. Shi, "Comparison between genetic algorithms and particle swarm optimization," in *Evolutionary programming*, V.W. Porto, N. Saravanan, D. Waagen, and A. E. Eiben (Eds.), 1998.
[23] Y. Shi, and R.C. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the IEEE International Conference on Evolutionary Computation*, Piscataway, NJ: IEEE Press. 1998, pp. 69-73.
[24] Y. Shi and R.C. Eberhart, "Empirical study of particle swarm optimization," in *Proceedings of the 1999 Congress on Evolutionary Computation*, Piscataway, NJ: IEEE Service Center, 1999, pp. 1945–1950..
[25] Y. Shi and R.C. Eberhart, "Particle Swarm Optimization with Fuzzy Adaptive Inertia Weight", in *Proceedings of the Workshop on Particle Swarm Optimization*, Indianapolis, IN: Purdue School of Engineering and Technology, IUPUI press, 2011.
[26] J. Hadamard, *Sur les problems aux derivees partielles et leur signification physique,* Princeton University Bulletin, Princeton, 1902.
[27] M. Servin, J.L. Marroquin and F.J. Cuevas, "Fringe-follower regularized phase tracker for demodulation of closed-fringe interferograms," *J. Opt. Soc. Am. A.*, vol. 18, pp. 689-695, 2001.
[28] M. Servin, J.L. Marroquin and F.J. Cuevas, "Demodulation of a single interferogram by use of a two-dimensional regularizad phase-tracking technique," *Appl. Opt.*, vol. 36, no. 19, pp.4540-4548, 1997.
[29] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on System, Man, and Cybernetics,* SMC vol. 9, no. 1, 1979.
[30] A. Glassner, "Fill 'Er Up!" *IEEE Computer Graphics and Applications*, vol. 21, no. 1, pp. 78-85, Jan./Feb. 2001.
[31] R.M Haralick and L.G. Shapiro, *Computer and Robot Vision*, Vol. 1, Addison-Wesley, 1992.
[32] A. Rosenfeld and J.L. Pfalz, "Sequential Operations in Digital Picture Processing," *Journal of the ACM*, vol. 13, no. 4, pp. 471-494, 1966.