Research Article

FSD-HSO Optimization Algorithm for Closed Fringes Interferogram Demodulation

Ulises H. Rodriguez-Marmolejo,[1,2] Miguel Mora-Gonzalez,[1] Jesus Muñoz-Maciel,[1] and Tania A. Ramirez-delreal[1,3]

[1]*Departamento de Ciencias Exactas y Tecnología, Centro Universitario de los Lagos, Universidad de Guadalajara, 47460 Lagos de Moreno, JAL, Mexico*
[2]*Departamento de Ingeniería Eléctrica y Electrónica, Instituto Tecnológico de Aguascalientes, 20256 Aguascalientes, AGS, Mexico*
[3]*Departamento de Posgrado e Investigación, Universidad Politécnica de Aguascalientes, 20342 Aguascalientes, AGS, Mexico*

Correspondence should be addressed to Miguel Mora-Gonzalez; mmora@culagos.udg.mx

Due to the physical nature of the interference phenomenon, extracting the phase of an interferogram is a known sinusoidal modulation problem. In order to solve this problem, a new hybrid mathematical optimization model for phase extraction is established. The combination of frequency guide sequential demodulation and harmony search optimization algorithms is used for demodulating closed fringes patterns in order to find the phase of interferogram applications. The proposed algorithm is tested in four sets of different synthetic interferograms, finding a range of average relative error in phase reconstructions of 0.14–0.39 rad. For reference, experimental results are compared with the genetic algorithm optimization technique, obtaining a reduction in the error up to 0.1448 rad. Finally, the proposed algorithm is compared with a very known demodulation algorithm, using a real interferogram, obtaining a relative error of 1.561 rad. Results are shown in patterns with complex fringes distribution.

1. Introduction

Interferometry includes a series of techniques used in the measurement of aberrations, deformations, flatness, and perturbations. It may also be applied to measure variables as temperature gradients, strain analysis, depth measurement, and so forth. Widely known techniques to demodulate interferograms are phase shifting algorithms for several images and the Fourier method for a single linear carrier fringe pattern [1]. However, these techniques are difficult to implement when the object under study changes fast and continuously, and the dynamic range of the phase does not allow the use of a large linear carrier without infringing the sampling theorem. In those situations a single interferogram with closed and possibly complex fringe distribution must be analyzed to recover the phase information related to the physical phenomena being measured. This is known as a difficult task since there are many solutions that are compatible with

the measured data but lacks physical meaning. The accuracy of measurements carried out from a single fringe pattern that includes closed fringes is thus intensely dependent on the phase distribution of the recorded interferogram being estimated. Recently, many phase recovering methods have been developed as combination of genetic algorithms and parametric methods [2–4], soft computing techniques applied to Zernike polynomials [5], combination of genetic algorithms and frequency guided sequential demodulation [6], particle swarm optimization [7], unwrapping of phase maps with sign changes [8], two-dimensional regularized phase-tracking technique [9], and so forth. In general, there are not particular processes that succeed in obtaining the phase for any given interferogram, but all of them are limited to specific features of the fringe pattern.

The phase demodulation trouble has been formulated as an optimization challenge, where soft computing procedures may be used to find the phase solution that best matches

the nonlinear equation represented by fringe patterns. Few years ago, genetic algorithms have been tested [2]; the authors developed a parametric method for fringe pattern demodulation using a genetic algorithm (GA). A parametric estimation of the coefficients of a 15th degree Zernike polynomial is used in order to approximate the phase; a population of chromosomes is programmed within the coefficients to calculate the phase. A cost function is then employed considering the number of the observed fringes and the fringes that result from the recovered phase match, the phase softness, and the prior knowledge of the object. Normally, the final solution of the GA is based on a cost function, which is stated as the comparison between the better individual in the population and the target (real fringes); a population evolution process is allowed until a cost function average threshold is achieved. The authors reported a root mean square (rms) error of 0.12 radians. This method was applied to noisy fringe patterns and to a single closed fringe image. Additional improvements and variations of this work were subsequently presented by the same research team [3, 4].

Another soft computing technique used for phase reconstruction is particle swarm optimization (PSO). This algorithm was introduced by Kennedy and Eberhart in 1995 [10], as an evolving optimization technique. In 2012 Jiménez et al. [7] used PSO for phase recovery; they compared a GA and a PSO for phase recovery on several fringe patterns, obtaining errors of 0.4281 and 0.313 rad., respectively, showing an improvement in accuracy of PSO over GA; processing time improvements were announced, but no results were shown.

As mentioned before, the demodulation of a single interferogram often involves a combination of methods (GA + Zernike, PSO + Zernike, Neuronal networks + others, etc.). In 2009 Wang and Kemao reported a new hybrid method; they used frequency guided sequential demodulation (FSD) as interferogram demodulator, combined with Levenberg Marquardt (LM) optimization [11], method implemented by their quickness and efficiency in fringes demodulation.

In this work a FSD with harmony search optimization (HSO) is investigated in order to test the performance in a single interferogram with closed fringes. The main motivation is the advantages of the HSO technique over other soft computing techniques already reported. The HSO technique was inspired in the observation of musical composition to search a perfect harmony and was introduced by Geem et al. in 2001 [12] and has found its way in several applications as diverse as engineering, math, industrial process, biology, and so forth [13–19]. An excellent recent review and categorization of the applications of HSO was conducted by Manjarres et al. in 2013 [20]. Some advantages of this method are that it uses simple algebraic equations and real values, while the derivative information is unnecessary unlike GA and other optimization techniques.

In the following section, the physical theory of the interferograms is presented as well as the concepts of the HSO and FSD algorithms. In the next section, the image-processing techniques and the experimental setup used to implement the soft computing proposed method are described. Finally, in the last two sections the results and the conclusions are presented, respectively.
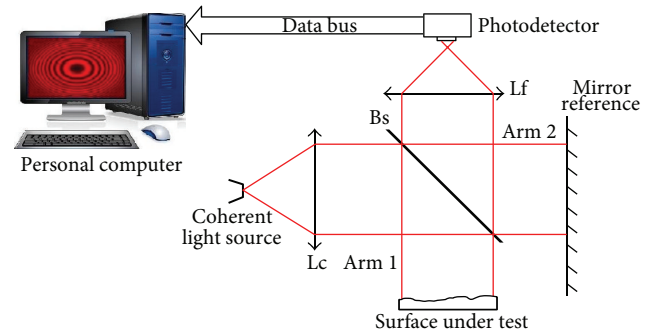

Figure 1: Interferogram acquisition system with a Twyman-Green interferometer.

2. Theory

Metrology has techniques such as fringe projection profilometry and optical interferometry to measure physical quantities in many areas of engineering and science, but the importance of these methods lies in the fact that they are noninvasive procedures [21]. Recently, advances in computational techniques have the potential to extend the measuring capabilities of optical metrology applications. In the present section the optical metrology basis, the harmony search optimization model, and the frequency guided sequential demodulation process are shown.

2.1. Optical Interferometry. Interferometry studies the engagement of two or more light waves, where one of them has suffered a modification by one characteristic of an object being tested [22]. Demodulation of the phase is the most important task in interferometry measurements; the phase is related to a physical quantity to be measured.

The optical arrangement, shown in Figure 1, is a Twyman-Green interferometer setup (a Michaelson interferometer modification). The interference is produced by the difference of optical path between the two arms of the interferometer. The interferogram is reordered by a photo detector array (e.g., a charged coupled display camera) and then digitized for show on a monitor or stored for further processing with computational algorithms like unwrapped phase, digital filtering, demodulation phase, and so forth [23]. The optical components Lc, Lf, and Bs are a positive collimating lens, a positive focusing lens, and a beam splinter, respectively. The fringe pattern intensity is modeled by

$$I(x, y) = a(x, y) + b(x, y) \cos(\varphi(x, y)), \quad (1)$$

where $I(x, y), a(x, y), b(x, y), \varphi(x, y)$, and (x, y) are the interference fringe pattern intensity, the background illumination, the modulation amplitude, the phase term, and the spatial coordinates of the surface under test, respectively.

2.2. Harmony Search Optimization. The harmony search is a metaheuristic optimization algorithm; it was created by Lee and Geem in 2005 [24]; it was inspired in music harmony and is a powerfully soft computing tool to solve many optimization problems. The HSO algorithm is structured in five steps
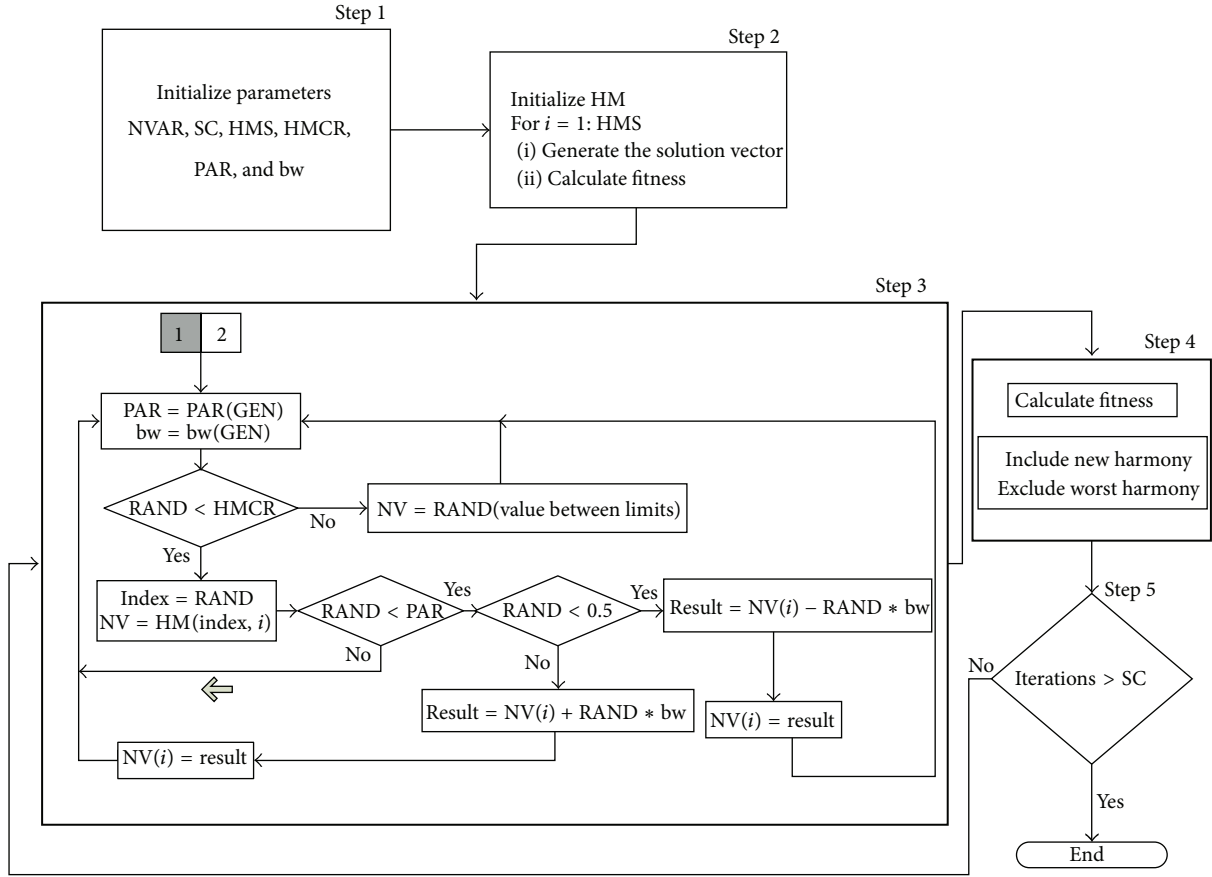
Figure 2: Flow diagram to indicate the structured HSO algorithm.

[25], as shown in Figure 2. Those steps are the following: (1) initialize the problem and algorithm parameters; (2) initialize the Harmony Memory (HM) array; (3) improvise a new harmony from HM; (4) update the HM; and (5) check the stopping criterion. These steps are described as follows.

Step 1. The optimization problem is specified and initialized its parameter values as follows:

$$\text{Mininize} \quad f(x)$$
$$\text{subject to} \quad x_i \in X_i = 1, 2, \ldots, \text{NVAR}, \tag{2}$$

where $f(x)$, x, and X_i are the objective function, the set of each decision variable x_i, and the set of the possible range of values for each decision variable, respectively. Also, the search range of X_i is $x_i^L \le X_i \le x_i^U$, with x_i^L and x_i^U as the lower and upper bounds for each decision variable, respectively. The parameters to initialize the algorithm are the number of decision variables (NVAR); the number of improvisations or stopping criterion (SC); the number of solution vectors in the Harmony Memory or the Harmony Memory Size (HMS); Harmony Memory Considering Rate (HMCR); Pitch Adjusting Rate (PAR) for each generation; and the arbitrary distance bandwidth for each generation (bw). In the following steps the role of some of these parameters is explained.

Step 2. The Harmony Memory (HM) array is complete with a random solution vectors as the HMS, as can be seen in the following:

$$\text{HM} = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_N^1 \\ x_1^2 & x_2^2 & \cdots & x_N^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{\text{HMS}} & x_2^{\text{HMS}} & \cdots & x_N^{\text{HMS}} \end{bmatrix}, \tag{3}$$

where HM acts as a dynamical memory address, where the sets of decision variables (all the solution vectors) are saved. This HM is similar to the genetic array in the genetic algorithm (GA) [26].

Step 3 (improvise a new harmony with parameters adjustment). For this improvisation, a new harmony vector $x' = (x_1', x_2', \ldots, x_N')$ is generated taking into account three basic considerations [27]: memory consideration, pitch adjustment, and random selection.

In the memory consideration, the value of the first decision variable x_1' for the new vector is chosen from any of the values in the specified HM. The values of other decision variables $x_2', \ldots, x_N'$ are chosen in the same manner. The

```
Generate the HM (initial harmonics as real number arrays)
Define the PAR
Define the HMCR
while (Max number of iterations > SC)
    Create new harmonics by accepting better harmonics
    Adjust PAR to take new solutions (harmonics)
        if (rand > HMCR)
            Choose an existing harmonic randomly
        else-if (rand > PAR)
            Adjust the pitch randomly within a bandwidth (bw)
        else
            Create new harmonics via randomization
        end if
    Accept the new solutions (harmonics) if better
end while
```

ALGORITHM 1: The improvisation procedure of HSO.

HMCR has a range of values from 0 and 1, while 1-HMCR is the possibility of creating a random HM member [27].

In the second consideration, a pitch adjusting is performed only after a value is chosen from the HM. The PAR considers the possibility of change of some elements of HM.

In the last consideration, a new vector x_i' is generated with a pitch adjustment threshold of 0.5, and the definition of this vector is described by

$$x_i' = x_i \pm \text{RAND}\,(\,) * \text{bw}, \tag{4}$$

where bw has an uniform distribution between -1 and 1. The function RAND( ) is fed with a random number between 0 and 1 to adjust the PAR. The sign of the addition in (4) depends of the value obtained with RAND function; if the value is less than 0.5 the sign is positive, and negative in other case.

Other parameters used in this step are NV and GEN, a new vector to improvise HM and for the depuration of the actual generation, respectively.

Step 4 (update HM). If the new harmony vector is better than the worst harmony vector in HM, then this latter is replaced by the new harmony vector. The HM is then sorted by the objective function value.

Step 5 (check the stopping criterion). If the SC is satisfied, computation is terminated. Otherwise, Steps 3 and 4 are repeated.

The HMCR and PAR parameters introduced in Step 3 help the algorithm to find locally and completed improved solutions, respectively. PAR and bw in HS algorithm are important parameters in fine-tuning of the optimized solution vectors and can be potentially useful in adjusting convergence rate of the algorithm to an optimal solution. The HSO pseudocode is shown in Algorithm 1.

The traditional HS algorithm uses fixed value for both PAR and bw. In the HS method PAR and bw values adjusted in initialization step (Step 1) and cannot be changed during new generations. Large bw values with small PAR values can produce poor performance of the algorithm and increased iterations are needed to find an optimum solution. Although in early generations bw must take a bigger value to enforce the algorithm to increase the diversity of solution vectors, small bw values in final generations increase the fine-tuning of solution vectors. Also, small bw values with large PAR values usually cause the improvement of best solutions in final generations in which the algorithm converges to the optimal solution vector [28].

2.3. Frequency Guided Sequential Demodulation. The frequency guided sequential demodulation (FSD) method was created by Kemao and Soon in 2007 [29], and it is used for recovery from the phase of closed fringe patterns. The algorithm of this method is explained in the following six sentences.

The first sentence is fringe denoising. The noise of (1) will be removed (low-pass filtering).

The second sentence is normalization. The parameters of resulting closed interferogram are normalized [30]; this is the background $a(x, y)$ and the amplitude $b(x, y)$; in effect, (1) is modified in order to find the interference fringe pattern normalized intensity

$$I_n(x, y) = \cos(\varphi(x, y)). \tag{5}$$

The third sentence is extracting the phase. The direct phase is extracted by the following equation:

$$\widehat{\varphi}_{wa}(x, y) = \cos^{-1}(I_n(x, y)) \in [0, \pi], \tag{6}$$

which is used to calculate the iteration frequencies or intermediate frequencies ξ_{xa} and ξ_{ya}; those frequencies are temporal, whilst they refine optimization in order to find the final frequencies ω_x and ω_y; this is, $(\omega_x, \omega_y) = \widehat{\omega}(x, y)$; it is feasible to use exhaustive search algorithm that guarantees a global minimum [31].

The fourth sentence is about the extraction of local frequency. The true phase is to be considered locally linear, as

$$\widetilde{\varphi}_a\left(x, y; u, v, \widetilde{\mathbf{p}}\right)$$
$$= \widehat{\varphi}_{wa}\left(x, y\right) + \xi_{xa} \cdot \left(x - u\right) + \xi_{ya} \cdot \left(y - v\right), \tag{7}$$

where $(u, v), (x, y), (\xi_{xa}, \xi_{ya})$ and the operator "$\cdot$" are the coordinates of the pixel under study, neighbors pixel coordinates, the internal frequencies while the optimization process is being performed, and the dot product, respectively. In other words $\widetilde{\mathbf{p}} = (\xi_{xa}, \xi_{ya})^T$ is the intermediate parameter vector. Then, a virtual fringe pattern can be generated as

$$\widetilde{I}\left(x, y; u, v, \widetilde{\mathbf{p}}\right) = \cos\left[\widetilde{\varphi}_a\left(x, y; u, v, \widetilde{\mathbf{p}}\right)\right]. \tag{8}$$

The fitness or energy cost function to be used by the HSO algorithm is equivalent to the squared difference between the virtual and the real fringe pattern; that is,

$$E\left(u, v; \widetilde{\mathbf{p}}\right) = \sum_{y=-\infty}^{\infty} \sum_{x=-\infty}^{\infty} \left[g\left(x - u, y - v\right) \right.$$
$$\left. \cdot \left[\widetilde{I}\left(x, y; u, v, \widetilde{\mathbf{p}}\right) - I_n\left(x, y\right)\right]^2 \right], \tag{9}$$

where $g(x, y)$ is a subimage window function briefly studied. The local frequency is estimated by minimizing the fitness function:

$$\widehat{\mathbf{p}}\left(u, v\right) = \arg \min_{\widetilde{\mathbf{p}}} E\left(u, v; \widetilde{\mathbf{p}}\right). \tag{10}$$

The fifth sentence is the extraction of estimated frequencies. Finally, the two coefficients found for the pixel under study or the estimated frequencies by HSO are

$$\widehat{\mathbf{p}}\left(u, v\right) = \left[\widehat{\omega}_{xa}\left(u, v\right), \widehat{\omega}_{ya}\left(u, v\right)\right]^T. \tag{11}$$

The sixth sentence is *eliminated the ambiguity error*. In order to determine the frequency guided sign, all subimages were processed to correct the problem of ambiguity sign. Palpably $\widehat{\mathbf{p}}(u, v)$ and $\widehat{\varphi}_{wa}(x, y)$ have sign ambiguity and the sign function $s(x, y)$ can be determined by forcing the local frequencies to be continuous; this is,

$$s\left(x_i, y_i\right) = \begin{cases} 1, & \text{if } \widehat{\omega}_a\left(x_i, y_i\right) \cdot \widehat{\omega}\left(x_{i-1}, y_{i-1}\right) \geq 0 \\ -1, & \text{if } \widehat{\omega}_a\left(x_i, y_i\right) \cdot \widehat{\omega}\left(x_{i-1}, y_{i-1}\right) < 0, \end{cases} \tag{12}$$

and the ambiguity is indicated by the subindex "a." The sign determination is then continued with a pixel that adjoins pixel (x_i, y_i) until all of the pixels have been processed. Once the sign field $s(x, y)$ is determined, both $\widehat{\omega}(x, y)$ and $\widehat{\varphi}(x, y)$ can be determined [31] by

$$\left[\widehat{\omega}\left(x, y\right), \widehat{\varphi}_w\left(x, y\right)\right]$$
$$= s\left(x, y\right)\left[\widehat{\omega}_a\left(x, y\right), \widehat{\varphi}_{wa}\left(x, y\right)\right]. \tag{13}$$

The Levenberg-Marquardt optimization is another algorithm that has been applied for this kind of recovery phase;

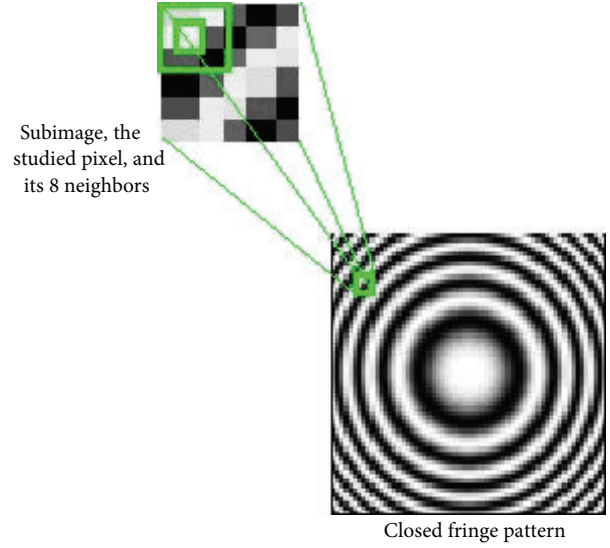Subimage, the studied pixel, and its 8 neighbors

Closed fringe pattern

FIGURE 3: Interferogram divided in 5×5-pixel subimages.

this algorithm uses the advantages of Gauss-Newton and gradient-descent methods based on adaptive rules [11].

3. Fringe Pattern Demodulation with FSD-HSO Algorithm

As explained above, (5) expresses the normalized interferogram and (6) represents the direct phase extracted. The process to demodulate a closed fringe pattern applying the FSD-HSO algorithm consists in splitting the complete closed fringe pattern into subimages of 5×5 pixels (see Figure 3). Additionally, a pixel is demodulated with the nearest eight neighbors, and the HSO algorithm is implemented after selecting the pixel in order to recover the phase for that element.

Further, the technique FSD is solved applying the HSO algorithm and taking into consideration (7), where the true phase is locally linear. Equation (11) expresses the two calculated frequencies by the HSO algorithm. Equation (8) describes the interferogram obtained with the estimated phase. Equation (9) depicts the fitness function for the HSO; it is the square of the subtraction of the real and the calculated fringe patterns; hence the proper frequencies cause an energy cost function minimization. The above process is applied in all pixels for the subimage. When the process is finished for all the subimages, (13) corrects the ambiguity error. Figure 4 shows the diagram of the demodulation process for each pixel applying HSO and FSD.

The pseudocode algorithms necessary to implement the proposed FSD-HSO procedure are shown in Algorithms 1, 2, 3, 4. First algorithm is the HSO method; second algorithm is the sign determination, in order to find the sign ambiguities of the estimate frequencies; third algorithm implements the procedure to clean the interferograms; fourth algorithm is the local frequencies calculation.
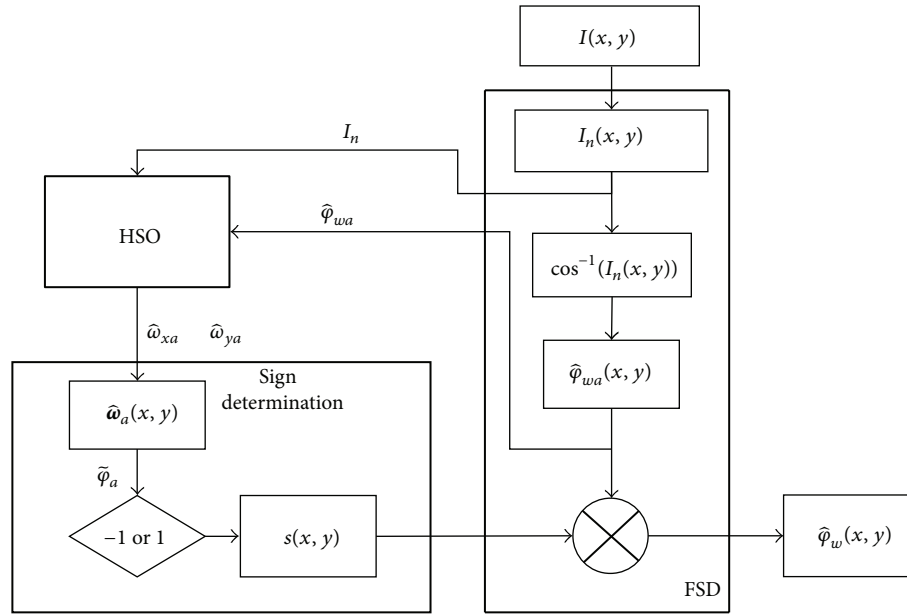
FIGURE 4: Demodulation process diagram applied to each pixel, using the proposed FSD-HSO algorithm.

---

for $x = 1, \ldots, n$
  for $y = 1, \ldots, m$
    Sign determination by coercing the continuity in the local frequencies:
      Calculate $s(x, y)$ through neighboring pixels (x_i, y_i)
      Calculate $[\widehat{\boldsymbol{\omega}}(x, y), \widehat{\varphi}_w(x, y)]$ with the correction of sign
  end of y
  end of x

ALGORITHM 2: Sign determination.

---

for $x = 1, \ldots, n$
  for $y = 1, \ldots, m$
    Fringe denoising (remove the noise without distorting the fringes)
    Background removal and amplitude normalization $I_n(x, y)$ between $[-1, 1]$
    Phase extraction directly by $\cos^{-1}(I_n(x, y))$
  end of y
  end of x

ALGORITHM 3: Fringe denoising.

---

for $x = 1, \ldots, n$
  for $y = 1, \ldots, m$
    for $u = 1, \ldots, 5$ (small window)
      for $v = 1, \ldots, 5$ (small window)
        Calculate the energy cost or fitness function, $E(u, v; \widetilde{\mathbf{p}})$
      end of v
    end of u
  end of y
  end of x
Minimizing the energy function, $\widehat{p}(u, v)$
Estimate the local frequencies by $[\widehat{\omega}_{xa}(u, v), \widehat{\omega}_{ya}(u, v)]^T$

ALGORITHM 4: Frequency extraction.

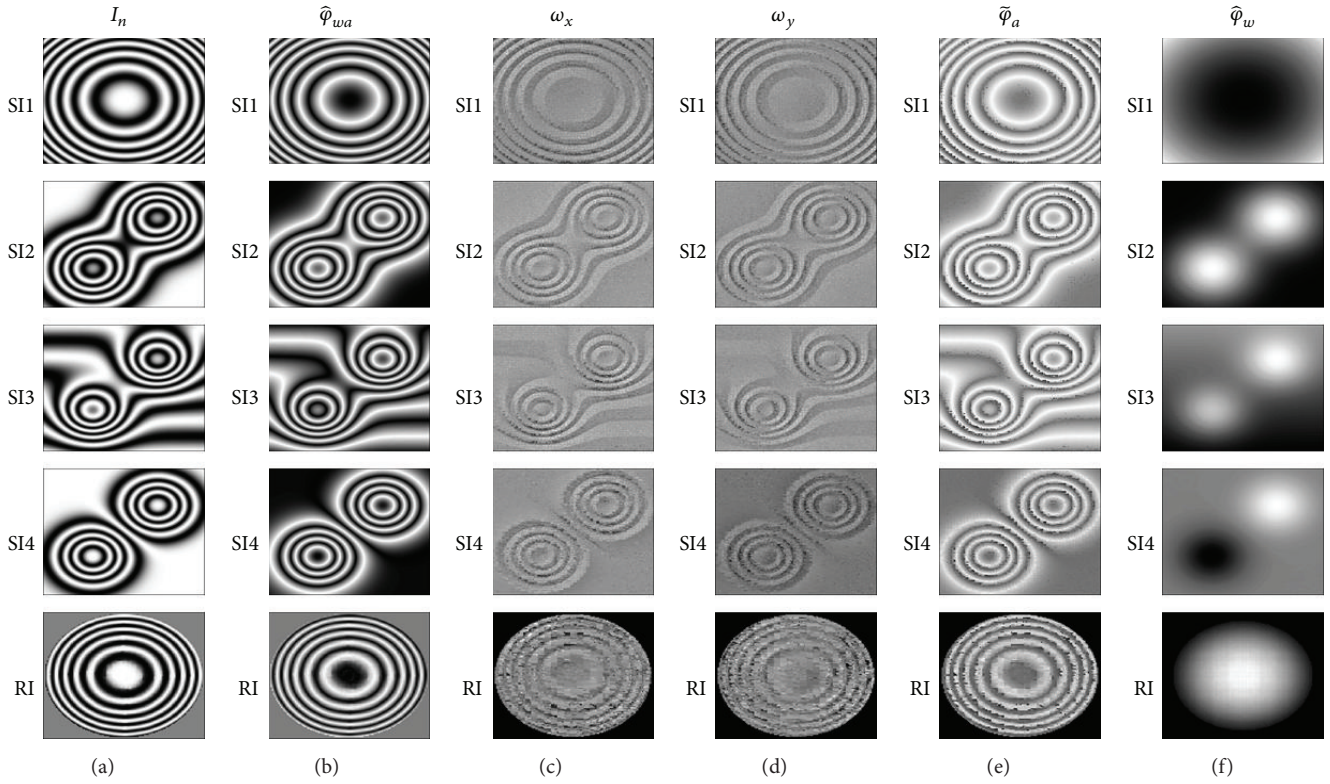| I_n | $\widehat{\varphi}_{wa}$ | ω_x | ω_y | $\overline{\varphi}_a$ | $\widehat{\varphi}_w$ |

FIGURE 5: FSD-HSO algorithm applied to demodulate synthetic and real interferograms: (a) normalized interferograms; (b) direct phases of interferograms; (c-d) calculated frequencies along (x, y) axis, respectively; (e) demodulated phases with ambiguity errors; and (f) correct demodulated phases without ambiguity error.

4. Results of Closed Fringe Pattern Demodulation with FSD-HSO Algorithm

A variety of physical quantities can be measured through the phase; it can be using interferometric techniques. The method to calculate the phase for a closed fringe pattern is explained below. Synthetic interferograms (closed fringe) are demodulated using the FSD-HSO algorithm. The physical variable associated with the obtained phase is smooth and continuous.

The HSO algorithm begins its search with HM and in this case is created by n rows of individuals (HMS $= n$) with two elements (NVAR), each with random values that correspond to the beginning of estimated frequencies by (14); this is

$$
\text{HM} = \begin{bmatrix} \omega_x^1 & \omega_y^1 \\ \omega_x^2 & \omega_y^2 \\ \omega_x^3 & \omega_y^3 \\ \vdots & \vdots \\ \omega_x^{\text{HMS}} & \omega_y^{\text{HMS}} \end{bmatrix}. \tag{14}
$$

The values of each parameter in HSO (NVAR, SC, HMS, PAR, HMCR, and bw) are according to the interferogram to be demodulated. In this work, four distinct simulated closed

fringe patterns (synthetic interferograms, SIs) and a real interferogram (RI) are exposed. The first example, synthetic-interferogram-one (SI1), corresponds to Newton rings interferogram, that is, classical defocus interference generated in a Twyman-Green or a Fabry-Perot interferometer; the synthetic-interferogram-two (SI2) is a pair of symmetrical positive lobules; the synthetic-interferogram-three (SI3) is a pair of asymmetrical positive lobules; finally, the synthetic-interferogram-four (SI4) is a pair of inverse lobules; RI is a noisy interferogram with defocus aberration. These five examples are a compendium of different difficulties of closed fringe demodulation. The normalization of the SIs and RI were calculated and shown in Figure 5(a). The normalized interferogram $I_n(x, y)$ was partitioned into subimages N_{xy} of 5×5 pixels and also is considering the linear phase to be local. The direct phase is created by inverse cosine of normalized image; see Figure 5(b). Each subimage was processed and estimated the frequencies ω_x and ω_y, along x and y directions, respectively; see Figures 5(c) and 5(d). Equation (10) represents the cost function; it is decreased close to the selected pixel (x_1, y_1) and its eight neighbors. A small value in cost function implies good performance in the demodulation. The direct phase $\widehat{\varphi}_{wa}(x, y)$ was processed by the FSD-HSO algorithm described in flow diagram of Figure 4; as seen in Figure 5(e), the sign ambiguity creates discontinuities in this step of phase reconstruction process.
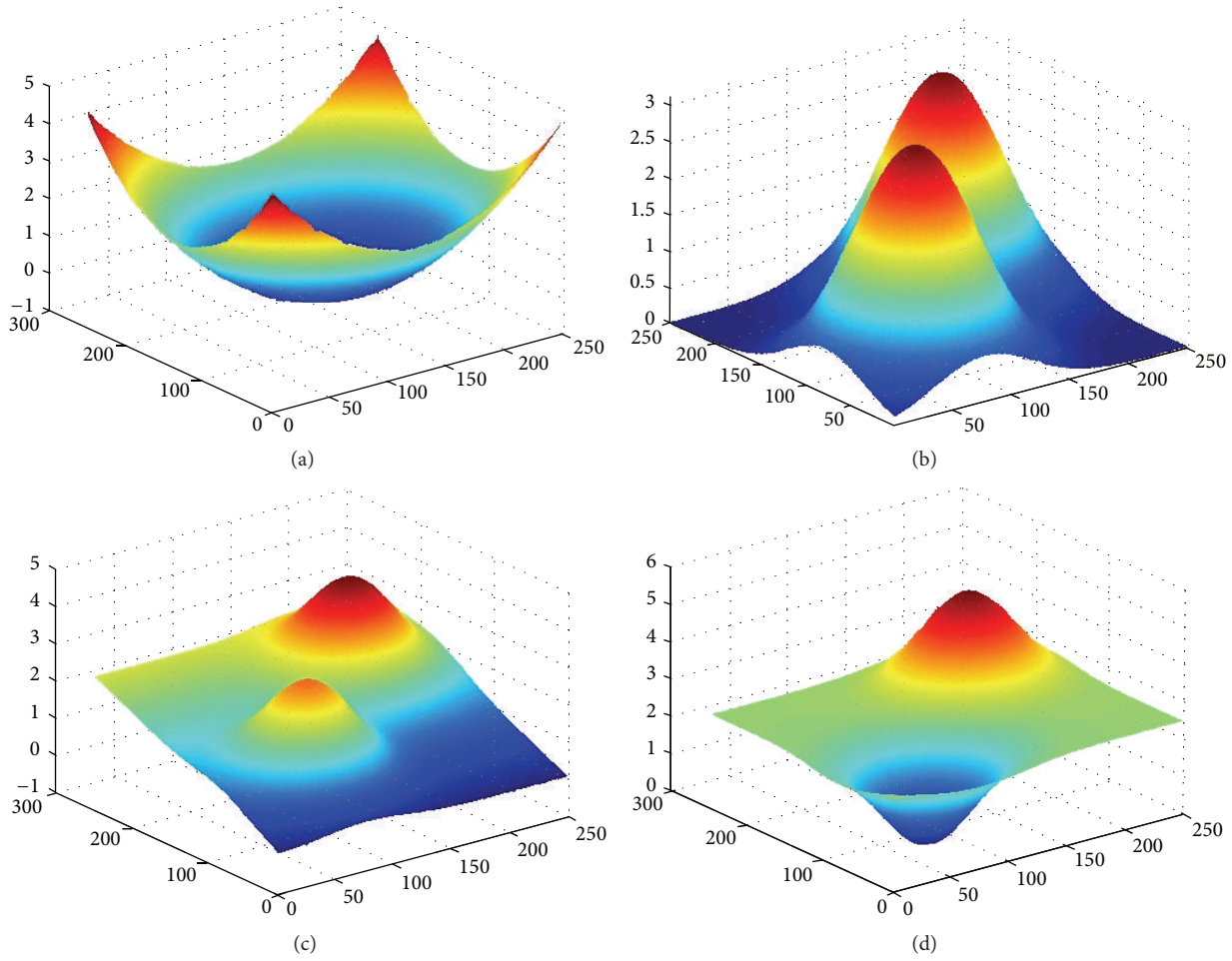
FIGURE 6: 3D estimated phases of SIs demodulations: (a) $\widehat{\varphi}_w$ of SI1; (b) $\widehat{\varphi}_w$ of SI2; (c) $\widehat{\varphi}_w$ of SI3; and (d) $\widehat{\varphi}_w$ of SI4.

In Figure 4 a block of sign determination was included because of the phase ambiguity problem presented in the function cosine; this is $\cos(\varphi_e(x, y)) = \cos(-\varphi_e(x, y))$; details of this method can be found in [31]. Once the subimage is completed, the HSO starts with other subimages; when the last subimage is demodulated, the procedure finishes. At last, the estimated phases $\widehat{\varphi}_w(x, y)$ are shown in Figure 5(f).

In order to observe minutiae details of the synthetically estimated phases, a series of 3D graphs are presented in Figure 6. The estimated phase of SI1 is observed as a single negative paraboloid, as shown in Figure 6(a); in counterpart, the other three SIs estimated phases have two peaks. The estimated phase of SI2 is a pair of positive symmetrical peaks; see Figure 6(b). However, the respective SI3 estimated phase, is observed as two positive asymmetrical peaks; see Figure 6(c). At last, two inverse peaks are represented in Figure 6(d) as SI4 estimated phase. This last interferogram was the most difficult to demodulate, because the search process of the HSO algorithm was slow due to the increase in iterations for the PAR and bw adjustment. It is seen that the combination of the HSO and FSD methods can handle interferograms with more complex fringe distribution than circular fringes.

The average relative error was calculated by the following equation [3]:

% error

$$= \frac{100}{MN} \sum_x \sum_y \left| \frac{\varphi_o(x, y) - \widehat{\varphi}_w(x, y)}{\max(\widehat{\varphi}_w(x, y)) - \min(\widehat{\varphi}_w(x, y))} \right|, \quad (15)$$

where φ_o, MN are the original interferogram phase and the interferogram size, respectively.

In addition to the synthetic analysis, the RI estimated phase demodulated with FSD-HSO was compared with a robust unwrapped phase method, implemented by [8], for this occasion that method is taken to calculate the original interferogram phase. In order to compare the two results visually, their 3D phases are shown in Figures 7(a) and 7(b); these two graphs are very similar; to complement the analysis, the variance between the original phase and the calculated phase $|\varphi_o(x, y) - \widehat{\varphi}_w(x, y)|$ was obtained, for this RI whose difference was very small (around ~0.5 radians), as shown in Figure 7(c). For SI analysis, the original phase is the phase with which was programmed each synthetic interferogram.
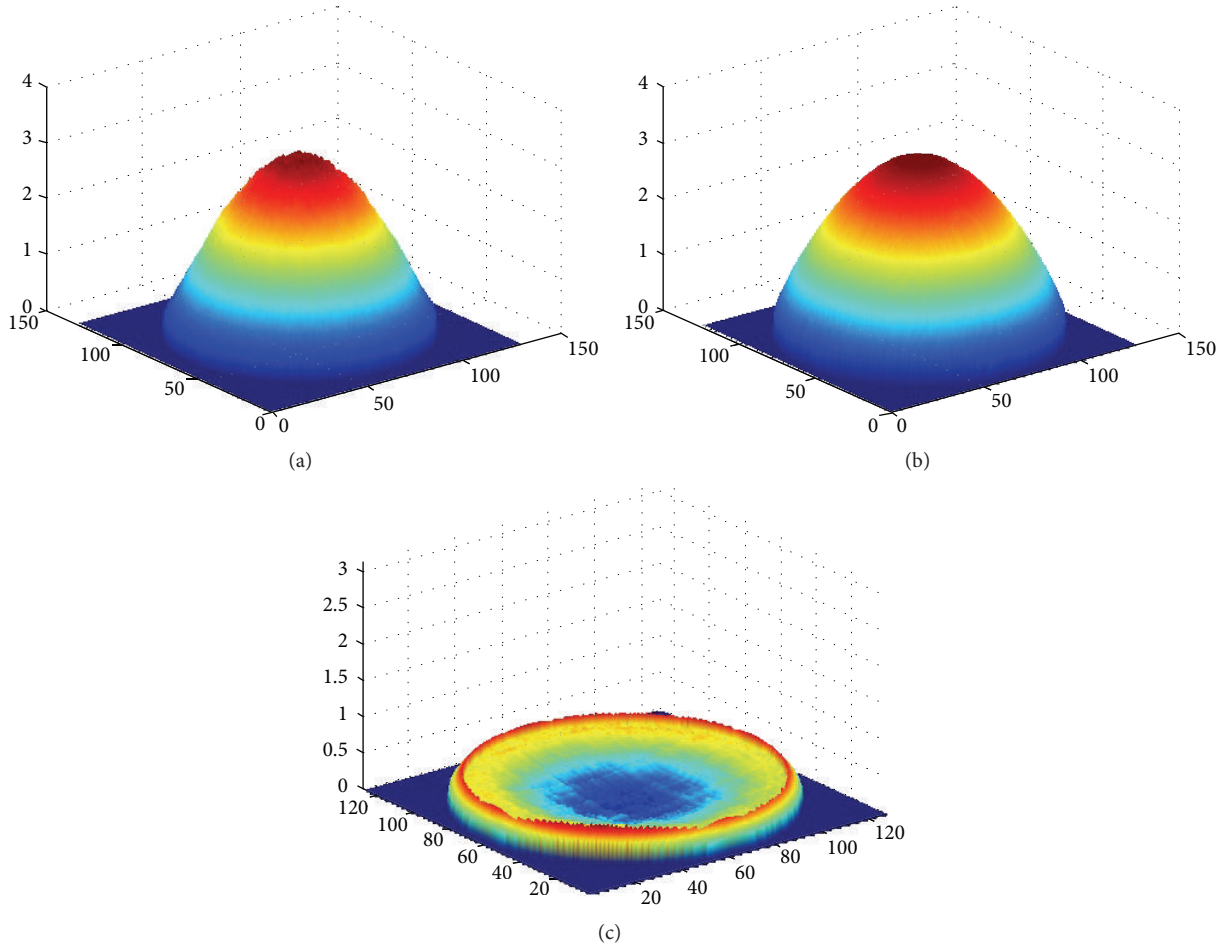
FIGURE 7: 3D analysis of RI demodulation phase: (a) $\widehat{\varphi}_w$ of RI with FSD-HSO demodulation; (b) φ_0 of RI with unwrapped phase method; and (c) phases difference $|\varphi_0 - \widehat{\varphi}_w|$.

TABLE 1: Comparison between GA and HSO optimization techniques. The demodulation method used in both cases was FSD.

Interferogram	p-v deviation (radians)		rms deviation (radians)		% error (radians)		Time processing (seconds)	
	AG	HSO	AG	HSO	AG	HSO	AG	HSO
SI1	4.8740	4.8639	1.9526	1.9009	1.1729	0.1700	398.63	234.50
SI2	3.0328	3.0031	1.3571	1.3043	2.1474	0.2123	444.83	247.13
SI3	4.3100	4.3090	2.1649	2.1110	1.3505	0.1448	407.88	239.93
SI4	4.9980	4.9980	2.7042	2.6344	1.2058	0.3984	5412.98	3026.59
RI	3.485	3.1491	1.346	1.3471	5.2806	1.5610	192.59	87.54

Additionally, the results of HSO algorithm are compared with genetic algorithm (GA) method, both using FSD as demodulation technique. Results of this comparison can be seen in Table 1, where column one contains the five different interferograms used in the present work. The second column of Table 1 contains the peak to valley (p-v) deviations, where column represents the maximal and minimal values obtained in the respective estimated phases. The third column of the same table presents the rms deviations of the estimated phases. In p-v and rms calculus significant differences cannot be observed between GA and HSO methods, but these data

are presented because of their physical significance. The average relative errors were presented in fourth column; it can be observed that HSO algorithm has smaller errors than GA algorithm; the difficulties presented in SI4 and RI (pronounced sign peaks asymmetry and/or very noisy image) increment a little the error in both methods; the biggest error was 1.561 rad.; therefore, the robustness of FSD-HSO is confirmed. The last column presents the time processing, where the FSD-HSO demonstrated as a fast method with symmetrical sign peaks interferograms demodulation (until 1.5 minutes), but the time processing increases as different
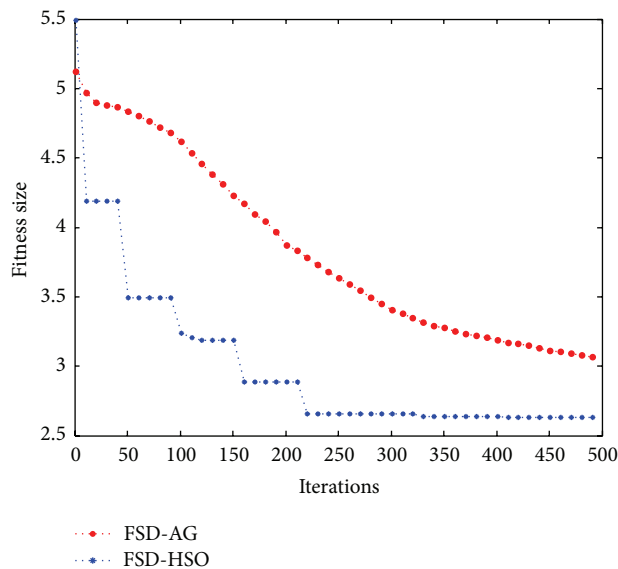
FIGURE 8: Convergence comparison for SI1 interferogram pixels.

TABLE 2: Parameters used in FSD-HSO optimization algorithm.

Interferogram	NVAR	SC	HMS	HMCR	PAR	bw	Limits
SI1	2	800	1000	0.9	0.75	0.05	$[-0.3, 0.3]$
SI2	2	1000	1500	0.9	0.7	0.08	$[-0.2, 0.2]$
SI3	2	1000	1800	0.85	0.75	0.07	$[-0.5, 0.5]$
SI4	2	1500	2200	0.9	0.9	0.1	$[-0.5, 0.5]$
RI	2	1500	2000	0.85	0.7	0.09	$[-0.5, 0.5]$

sign peak lobules appear in the interferogram. Results of Table 1 are supported by the plot of Figure 8, in which the number of iterations for the proposed method is about one half to achieve convergence.

The selected parameters used in this work are shown in Table 2. The entire SIs and RI images had MN size of 250×250 pixels and 128×128 pixels, respectively. In order to obtain consistent results, all the calculus is made with the same computational system: Intel Celeron B815 1.6 GHz, with 2 GB of RAM, 32-bit operative system, Windows 7, and Matlab R2011a 7.120.635. The FSD-GA at its best operation, used to compare the proposed algorithm, has the following parameters: individuals 1500–2500; bits by gen 12–15; gens 2; generations 1000–1500; limits of each coefficient $[-0.3, 0.3]$–$[-0.5, 0.5]$.

Experimentally it is observed that the proposed algorithm's tuning parameters depend on the interferogram characteristics. The best resolution in the image demodulation was found experimentally where PAR must approach its upper limit (one) and bw tends to its lower limit (zero).

5. Conclusions

The use of FSD algorithm to demodulate the phase is a simple technique of soft computing, because just two variables for this work, ω_x and ω_y, need to be calculated compared to other polynomial approaches that calculate more variables or coefficients.

Instead, the HSO is a relatively new optimization technique that requires a few input parameters. The fitness function used by the HSO is small and the data search is relational, so convergence is fast and has a low percentage of error.

Therefore, the combined algorithm FSD-HSO is an uncomplex algorithm because of its algebraic nature, and, used in the demodulation of closed fringe patterns, its convergence is fast. The error calculated in synthetic interferograms simulated was 0.14–0.39 which is a very good approximation. The method is limited only to demodulation of closed fringe patterns.

The proposed algorithm is relatively fast compared to the FSD-AG algorithm; it converges with less iterations. It only presents difficulties in interferograms with different sign peak lobules, but this disadvantage is present in both methods.

Other methods to demodulation also shed low error rates, such as the AG-Zernike algorithm, but, using these two methods for example, the algorithm becomes very complex, since it is necessary to perform a lot of computational processes such as data mutations, data changes from binary to real, and a lot of variables to be estimated (a minimum of 7 variables).

Competing Interests

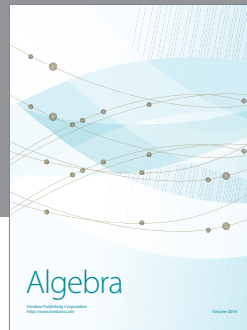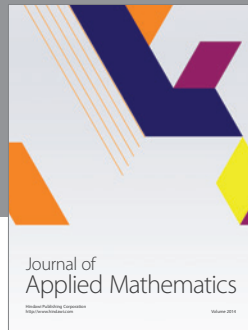The authors declare that they have no competing interests.

Acknowledgments

References

[1] D. Malacara, M. Servin, and Z. Malacara, *Interferogram Analysis for Optical Testing*, CRC Press, Boca Raton, Fla, USA, 2nd edition, 2005.

[2] 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," *Optics Communications*, vol. 203, no. 3–6, pp. 213–223, 2002.

[3] F. J. Cuevas, F. Mendoza, M. Servin, and J. H. Sossa-Azuela, "Window fringe pattern demodulation by multi-functional fitting using a genetic algorithm," *Optics Communications*, vol. 261, no. 2, pp. 231–239, 2006.

[4] L. E. Toledo and F. J. Cuevas, "Optical metrology by fringe processing on independent windows using a genetic algorithm," *Experimental Mechanics*, vol. 48, no. 4, pp. 559–569, 2008.

[5] L. E. Mancilla Espinosa, J. M. Carpio Valadez, 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.

[6] U. H. Rodriguez-Marmolejo, T. A. Ramirez-delreal, J. Muñoz-Maciel, and M. Mora-Gonzalez, "Combination of genetic algorithms and FSD applied to fringe pattern demodulation," in

Proceedings of the Image Reconstruction from Incomplete Data VIII, vol. 9600 of *Proceedings of SPIE*, 7 pages, San Diego, Calif, USA, August 2015.

[7] J. Jiménez, H. Sossa, F. Cuevas, and L. Gómez, "Demodulation of interferograms based on particle swarm optimization," *Polibits*, vol. 45, pp. 83–91, 2012.

[8] J. Muñoz-Maciel, F. J. Casillas-Rodríguez, M. Mora-González, F. G. Peña-Lecona, V. M. Duran-Ramírez, and G. Gómez-Rosas, "Phase recovery from a single interferogram with closed fringes by phase unwrapping," *Applied Optics*, vol. 50, no. 1, pp. 22–27, 2011.

[9] M. Servin, J. L. Marroquin, and F. J. Cuevas, "Demodulation of a single interferogram by use of a two-dimensional regularized phase-tracking technique," *Applied Optics*, vol. 36, no. 19, pp. 4540–4548, 1997.

[10] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, Perth, Australia, November-December 1995.

[11] H. Wang and Q. Kemao, "Frequency guided methods for demodulation of a single fringe pattern," *Optics Express*, vol. 17, no. 17, pp. 15118–15127, 2009.

[12] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation*, vol. 76, no. 2, pp. 60–68, 2001.

[13] G. Ingram and T. Zhang, "Overview of applications and developments in the harmony search algorithm," in *Music-Inspired Harmony Search Algorithm*, Z. W. Geem, Ed., vol. 191 of *Studies in Computational Intelligence*, pp. 15–37, Springer, New York, NY, USA, 2009.

[14] F. Deng, S. Tuo, L. Yong, and T. Zhou, "Construction example for algebra system using harmony search algorithm," *Mathematical Problems in Engineering*, vol. 2015, Article ID 836925, 15 pages, 2015.

[15] Q.-K. Pan, P. N. Suganthan, J. J. Liang, and M. F. Tasgetiren, "A local-best harmony search algorithm with dynamic subpopulations," *Engineering Optimization*, vol. 42, no. 2, pp. 101–117, 2010.

[16] M. Sulaiman, A. Salhi, B. I. Selamoglu, and O. B. Kirikchi, "A plant propagation algorithm for constrained engineering optimisation problems," *Mathematical Problems in Engineering*, vol. 2014, Article ID 627416, 10 pages, 2014.

[17] G. Li and Q. Wang, "A cooperative harmony search algorithm for function optimization," *Mathematical Problems in Engineering*, vol. 2014, Article ID 587820, 12 pages, 2014.

[18] M. Jaberipour and E. Khorram, "A new harmony search algorithm for solving mixed-discrete engineering optimization problems," *Engineering Optimization*, vol. 43, no. 5, pp. 507–523, 2011.

[19] X. Z. Gao, V. Govindasamy, H. Xu, X. Wang, and K. Zenger, "Harmony search method: theory and applications," *Computational Intelligence and Neuroscience*, vol. 2015, Article ID 258491, 10 pages, 2015.

[20] D. Manjarres, I. Landa-Torres, S. Gil-Lopez et al., "A survey on applications of the harmony search algorithm," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 8, pp. 1818–1831, 2013.

[21] K. J. Gassvik, *Optical Metrology*, John Wiley & Sons, New York, NY, USA, 1987.

[22] E. Hecht, *Optics*, Addison-Wesley, San Francisco, Calif, USA, 4th edition, 2002.

[23] D. Malacara, "Twyman-Green interferometer," in *Optical Shop Testing*, D. Malacara, Ed., chapter 2, pp. 46–96, John Wiley & Sons, Hoboken, NJ, USA, 3rd edition, 2007.

[24] K. S. Lee and Z. W. Geem, "A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice," *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 36–38, pp. 3902–3933, 2005.

[25] X. Z. Gao, X. Wang, and S. J. Ovaska, "Harmony search methods for multi-modal and constrained optimization," in *Music-Inspired Harmony Search Algorithm: Theory and Applications*, vol. 191 of *Studies in Computational Intelligence*, pp. 39–51, Springer, Berlin, Germany, 2009.

[26] Y. Xin-She, *Engineering Optimization. An Introduction with Metaheuristic Applications*, John Wiley & Sons, Hoboken, NJ, USA, 2010.

[27] L. dos Santos Coelho and D. L. A. Bernert, "A harmony search approach using exponential probability distribution applied to fuzzy logic control optimization," in *Recent Advances in Harmony Search Algorithm*, Z. W. Geem, Ed., vol. 270 of *Studies in Computational Intelligence*, pp. 77–88, Springer, 2010.

[28] S. Tuo, L. Yong, and T. Zhou, "An improved harmony search based on teaching-learning strategy for unconstrained optimization problems," *Mathematical Problems in Engineering*, vol. 2013, Article ID 413565, 29 pages, 2013.

[29] Q. Kemao and S. H. Soon, "Sequential demodulation of a single fringe pattern guided by local frequencies," *Optics Letters*, vol. 32, no. 2, pp. 127–129, 2007.

[30] J. A. Quiroga, J. Antonio Gómez-Pedrero, and Á. García-Botella, "Algorithm for fringe pattern normalization," *Optics Communications*, vol. 197, no. 1–3, pp. 43–51, 2001.

[31] Q. Kemao, *Windowed Fringe Pattern Analysis*, SPIE Press, Bellingham, Wash, USA, 2013.