

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Fringe Pattern Demodulation Using Evolutionary Algorithms

L. E. Toledo¹, F. J. Cuevas¹, J.F. Jimenez Vielma² and J. H. Sossa²

¹*Centro de Investigaciones en Optica A.C.,
Dept. of Computer Vision and Artificial Intelligence, Optical Division, Leon,*

²*Center for Computing Research, National Polytechnical Institute,
Artificial Intelligence Laboratory,
Mexico*

1. Introduction

Interferometers are used in metrology to measure temperature, displacement, stress and other physical variables. A typical interferometer split a laser beam using a beam divisor. Beam A is called reference, and is projected directly over a film or a CCD camera using mirrors or fiber optic. Beam B interact with the physical phenomenon to be measured. The interaction modifies the optic path of beam B; then it is projected over the same film or CCD camera that beam A. The total irradiance is modelled on eq. 1.

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

The information about the measure is embodied on an interferogram, that is, a fringe pattern image. In optical metrology, a fringe pattern carries information embedded in its phase, that represents the difference in optical path between beam A and beam B. 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, and $\varphi(x,y)$ is the phase term related to the physical quantity being measured. Figure 1 shows an interferogram and its associated phase $\varphi(x,y)$.



Fig. 1. Fringe pattern(a) and its phase map (b).

The problem is to recover the phase map from the fringe pattern image. The demodulation process can be achieved by different methods, depending on the characteristics of the fringe pattern. If the fringe pattern, or interferogram has open fringes (see fig. 2c) by adding a carrier or tilt onto the phase, phase is obtained using Takeda's Fourier Transform method (Takeda & Kobayashi, 1982) or the Phase Locked Loop (Servin & Rodriguez Vera, 1993). A carrier is a phase that increases or decreases linearly with x and or y .

PLL and Fourier methods can not be used if the interferogram has closed fringes or is not normalized. A normalized fringe pattern means $a(x,y) \approx 0$ and $b(x,y) \approx 1$ (see fig 3). Many methods can be used to normalize a fringe pattern (Quiroga et al, 2001). An interferogram can be normalized due to a tilt in the plane waves of beams, defocus, speckle noise, etc.

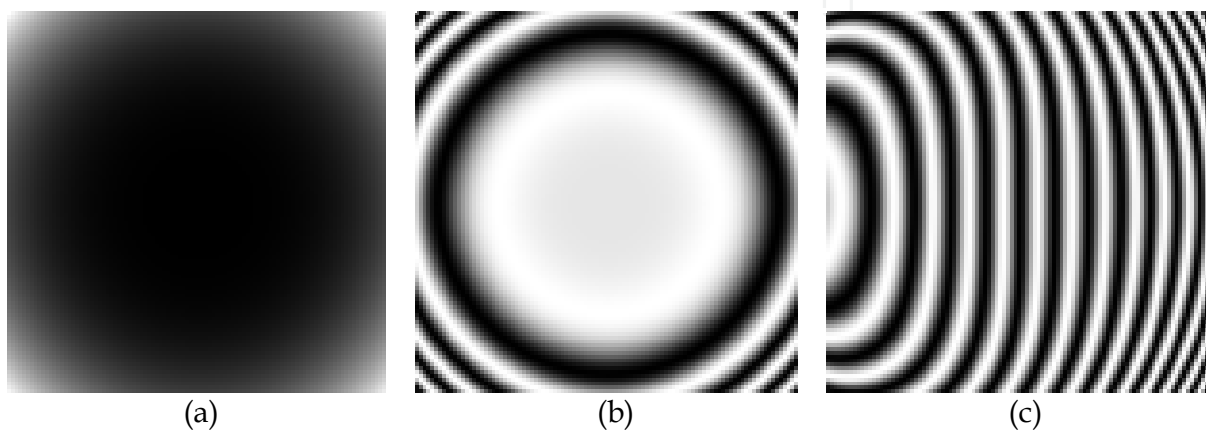


Fig. 2. Original phase (a), fringe pattern without carrier; (b), fringe pattern by adding a carrier (c).

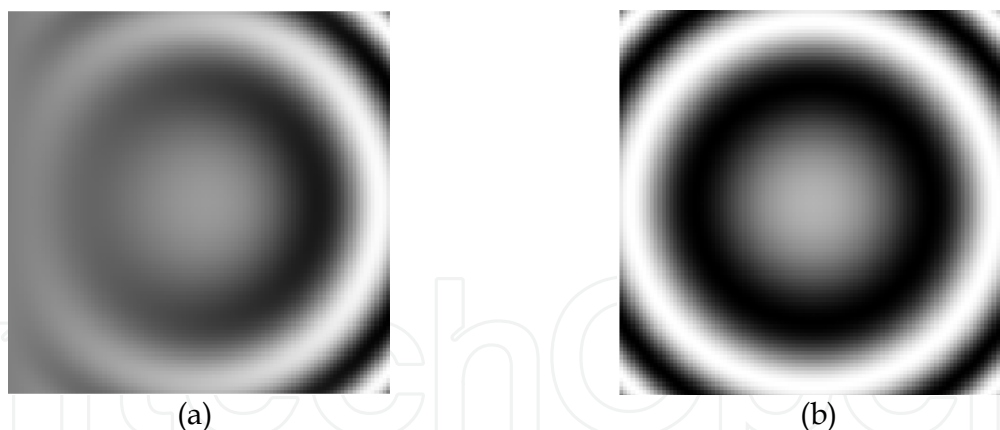


Fig. 3. (a) Non-normalized fringe pattern. $a(x,y) = 0.001y$, $b(x,y) = 2x$; (b) Normalized fringe pattern.

If it is possible to take three or more images and add a constant on the phase, a constant shift that is different for each fringe pattern (see fig. 4), phase shifting techniques are suitable (Malacara et al, 1998). It is not necessary to normalize the fringe pattern, but it is assumed that $a_1(x,y) \approx a_2(x,y) \approx a_3(x,y)$ and $b_1(x,y) \approx b_2(x,y) \approx b_3(x,y)$.

A drawback to phase shifted method is the real phase are not obtained, but a $\text{mod } 2\pi$ of the phase (fig. 5a). An unwrapping method (Ghiglia & Romero, 1994) is necessary to obtain the real phase (fig. 4b). The fringe patterns used on phase shifting should be normalized.

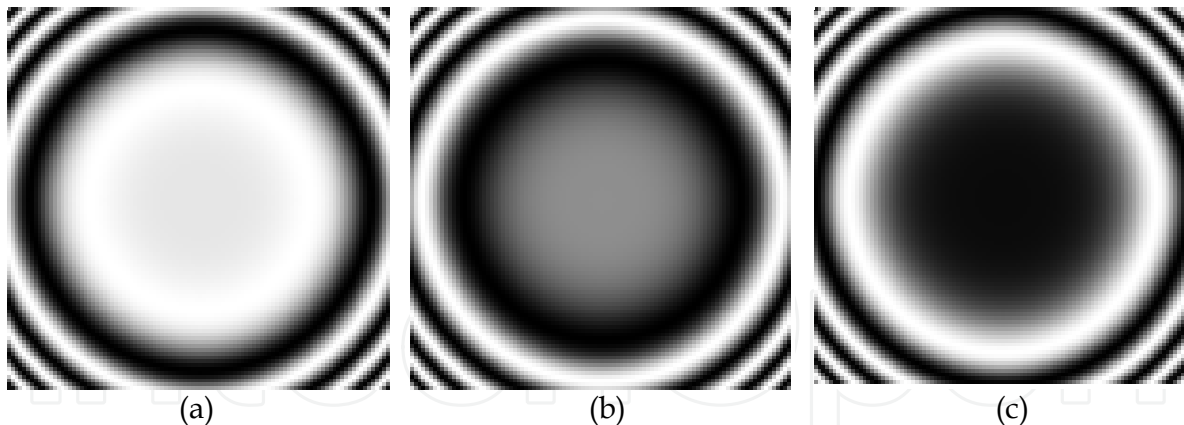


Fig. 4. (a) Original fringe pattern I; (b) Adding a constant phase of 120 degrees, I₂; (c) Adding a phase of -120 degrees, I₃.

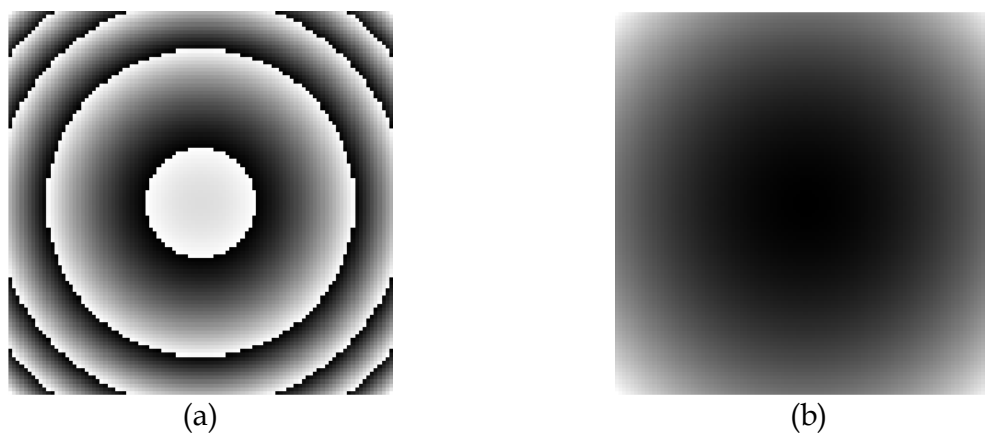


Fig. 5. (a) Wrapped phase; (b) Unwrapped phase.

Methods like the Phase Tracker (Servin et al, 2001a) and the Two-dimensional Hilbert Transform (Larkin et al, 2001) are used for closed fringes, normalized images. These methods are robust again a large amount of noise, but a subjacent condition is to fulfil Nyquist condition. Phase tracker gives an unwrapped phase so there is not necessary to use an unwrapping method. The phase tracker and Hilbert transform proposed a cost function that depends of some measure of the difference between the real phase and the estimated phase. The real phase is unknown so the original interferogram is used and compared to the fringe pattern of the proposed phase. More terms are added to introduce restrictions.

A problem with minimize a cost function is the danger of fall in a local minimum, far away from the optimal point. It is also possible to use soft computing algorithms, such as neural networks and evolutionary algorithms (EA). In the neural network technique, a multilayer neural network (MLNN) is trained by using fringe patterns, and the phase gradients associated with them, from calibrated objects (Cuevas et al, 2000); after the training, the MLNN can estimate the phase gradient when the fringe pattern is presented in the MLNN input. A genetic algorithm (GA) is a particular type of EAs. GA's are optimization algorithms that simulate natural evolution (Holland, 1975), and whereas GAs do not search for the best solution to a given problem, they can discover highly precise functional solutions and are very useful for nonlinear optimization problems or in the presence of

multiple minimums (Goldberg, 1989), where classic techniques like gradient descent, deterministic hill climbing or random search (with no heredity) fail.

Methods using GA (Cuevas et al, 2002), approximate the phase through the estimation of parametric functions. The chosen functions could be Bessel in the case of having fringes from a vibrating plate experiment, or Zernike polynomials, in the case of an optical testing experiment, and when not much information is known about the experiment, a set of low degree polynomials $p(\mathbf{a},x,y)$ can be used. A complicated pattern is demodulated dividing it into a set of partially overlapping windows fitting a low dimensional polynomial function in each window, so that no further unwrapping is needed (Cuevas et al, 2006).

2. Genetic algorithms

Genetic Algorithms (GAs) (Bäck et al, 2000), are adaptive heuristic search algorithm premised on the evolutionary ideas of natural selection and genetic. The basic concept of GAs is designed to simulate processes in natural system necessary for evolution, specifically those that follow the principles first laid down by Charles Darwin of survival of the fittest. As such they represent an intelligent exploitation of a random search within a defined search space to solve a problem.

First pioneered by John Holland in the 60s, Genetic Algorithms has been widely studied, experimented and applied in many fields in engineering worlds. Not only does GAs provide an alternative method to solving problem, it consistently outperforms other traditional methods in most of the problems link. Many of the real world problems involved finding optimal parameters, which might prove difficult for traditional methods but ideal for GAs. However, because of its outstanding performance in optimization, GAs has been wrongly regarded as a function optimizer. In fact, there are many ways to view genetic algorithms.

In a genetic algorithm, a population of strings (called chromosomes or the genotype of the genome, fig. 6), which encode candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem, evolves toward better solutions. Traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population.

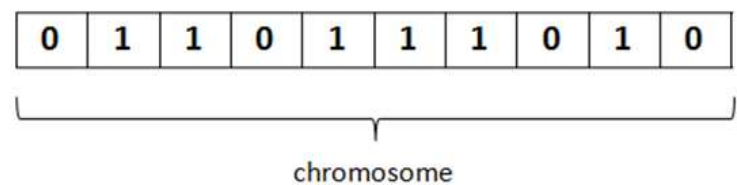


Fig. 6. A chromosome.

Genetic algorithms find application in bioinformatics, computational science, engineering, economics, chemistry, manufacturing, mathematics, physics and other fields.

A typical genetic algorithm requires:

1. A genetic representation of the solution domain, see fig. 7.
2. A fitness function to evaluate the solution domain.

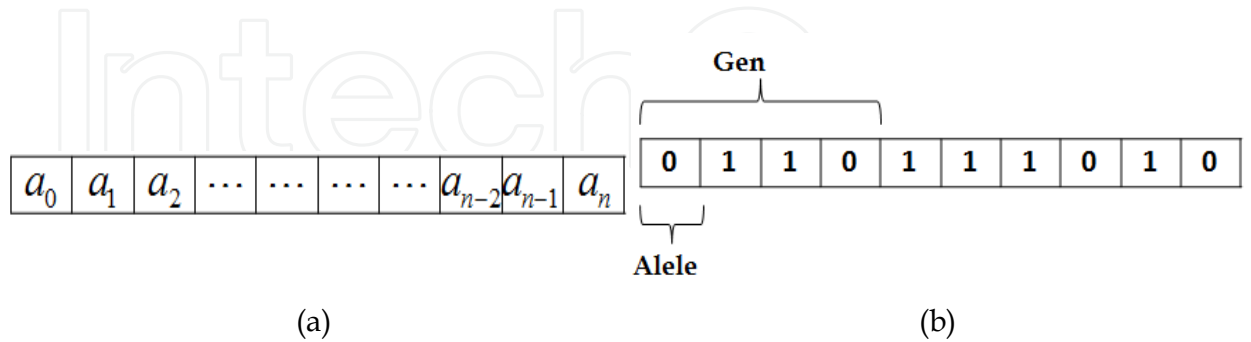


Fig. 7. (a) Representation of the solution domain; (b) Each gene is codified with a bit string.

A standard representation of the solution is as an array of bits. Arrays of other types and structures can be used in essentially the same way. The main property that makes these genetic representations convenient is that their parts are easily aligned due to their fixed size, which facilitates simple crossover operations. Variable length representations may also be used, but crossover implementation is more complex in this case.

The fitness function is defined over the genetic representation and measures the quality of the represented solution. The fitness function is always problem dependent. In some problems, it is hard or even impossible to define the fitness expression; in these cases, interactive genetic algorithms are used.

Once we have the genetic representation and the fitness function defined, GA proceeds to initialize a population of solutions randomly. Improve it through repetitive application of mutation, crossover, inversion and selection operators.

2.1 Initialization

At the first iteration many individual solutions are randomly generated to form the population. The population size depends on the nature of the problem, but typically contains several hundreds or thousands of possible solutions. Traditionally, the population is generated randomly, covering the entire range of possible solutions (the search space).

2.2 Selection

During each successive generation, a proportion of the existing population is selected to breed a new generation. Individual solutions are selected through a fitness-based process, where fitter solutions (as measured by a fitness function) are typically more likely to be selected. Certain selection methods rate the fitness of each solution and preferentially select the best solutions. Other methods rate only a random sample of the population, as this process may be very time-consuming.

A generic selection procedure may be implemented as follows:

1. The fitness function is evaluated for each individual, providing fitness values, which are then normalized. Normalization means dividing the fitness value of each individual by the sum of all fitness values, so that the sum of all resulting fitness values equals 1.
2. The population is sorted by descending fitness values.
3. Accumulated normalized fitness values are computed (the accumulated fitness value of an individual is the sum of its own fitness value plus the fitness values of all the previous individuals). The accumulated fitness of the last individual should be 1 (otherwise something went wrong in the normalization step).
4. A random number R between 0 and 1 is chosen.
5. The selected individual is the first one whose accumulated normalized value is greater than R .

Retaining the best individuals in a generation unchanged in the next generation, is called elitism or elitist selection. It is a successful (slight) variant of the general process of constructing a new population.

2.2.1 Roulette-wheel selection

Fitness proportionate selection, also known as roulette-wheel selection, is a genetic operator used in genetic algorithms for selecting potentially useful solutions for recombination.

In fitness proportionate selection, as in all selection methods, the fitness function assigns a value to possible solutions or chromosomes. This fitness level is used to associate a probability of selection with each individual chromosome. If f_i is the fitness of individual i

its probability of being selected is $p_i = \frac{f_i}{\sum_j^N f_j}$, where N is the number of individuals in the population.

This could be imagined similar to a Roulette wheel in a casino. Usually a proportion of the wheel is assigned to each of the possible selection based on their fitness value. This could be achieved by dividing the fitness of a selection by the total fitness of all the selections, thereby normalizing them to 1. Then a random selection is made similar to how the roulette wheel is rotated.

While candidate solutions with a higher fitness will be less likely to be eliminated, there is still a chance that they may be. Contrast this with a less sophisticated selection algorithm, such as truncation selection, which will eliminate a fixed percentage of the weakest candidates. With fitness proportionate selection there is a chance some weaker solutions may survive the selection process; this is an advantage, as though a solution may be weak, it may include some component which could prove useful following the recombination process.

The analogy to a roulette wheel can be envisaged by imagining a roulette wheel in which each candidate solution represents a pocket on the wheel; the size of the pockets are proportionate to the probability of selection of the solution. Selecting N chromosomes from the population is equivalent to playing N games on the roulette wheel, as each candidate is drawn independently.

2.2.2 Stochastic universal sampling

Stochastic universal sampling (SUS) is a technique used in genetic algorithms for selecting potentially useful solutions for recombination. It was introduced by James Baker.

SUS is a development of fitness proportionate selection which exhibits no bias and minimal spread. Where fitness proportionate selection chooses several solutions from the population by repeated random sampling, SUS uses a single random value to sample all of the solutions by choosing them at evenly spaced intervals.

While candidate solutions with a higher fitness will be less likely to be eliminated, there is still a chance that they may be. Contrast this with a less sophisticated selection algorithm, such as truncation selection, which will eliminate a fixed percentage of the weakest candidates. With fitness proportionate selection there is a chance some weaker solutions may survive the selection process; this is an advantage, as though a solution may be weak, it may include some component which could prove useful following the recombination process.

The analogy to a roulette wheel can be envisaged by imagining a roulette wheel in which each candidate solution represents a pocket on the wheel; the size of the pockets are proportionate to the probability of selection of the solution. Selecting N chromosomes from the population is equivalent to playing N games on the roulette wheel, as each candidate is drawn independently.

2.2.3 Tournament selection

It involves running several "tournaments" among a few individuals chosen at random from the population. The winner of each tournament (the one with the best fitness) is selected for crossover. Selection pressure is easily adjusted by changing the tournament size; if it is larger, weak individuals have a smaller chance to be selected.

Deterministic tournament selection selects the best individual (when $p = 1$) in any tournament. A *1-way* tournament ($k = 1$) selection is equivalent to random selection. The chosen individual can be removed from the population that the selection is made from if it is desired, otherwise individuals can be selected more than once for the next generation.

Tournament selection has several benefits: it is efficient to code, works on parallel architectures and allows the selection pressure to be easily adjusted.

2.3 Crossover

Crossover is a genetic operator used to vary the programming of a chromosome or chromosomes from one generation to the next. It is analogous to reproduction and biological crossover, upon which genetic algorithms are based. Cross over is a process of taking more than one parent solutions and producing a child solution from them.

Crossover techniques :

- One-point crossover (fig. 8a).
- Two-point crossover (fig. 8b).

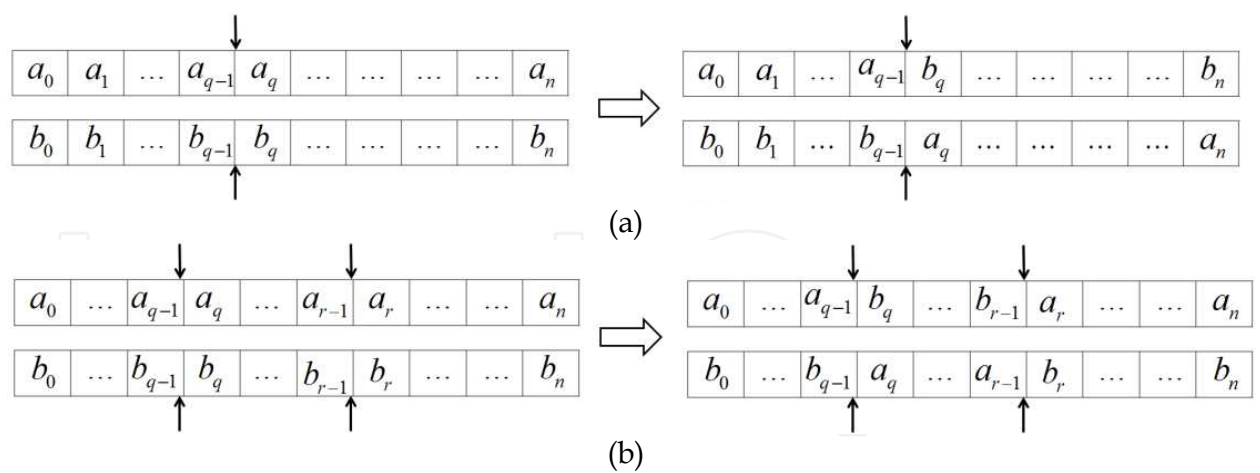


Fig. 8. (a) One point crossover; (b) Two point crossover.

2.4 Mutation

It is a genetic operator used to maintain genetic diversity from one generation of a population of algorithm chromosomes to the next. It is analogous to biological mutation. Mutation alters one or more gene values in a chromosome from its initial state. In mutation, the solution may change entirely from the previous solution. Hence GA can come to better solution by using mutation. Mutation occurs during evolution according to a user-definable mutation probability. This probability should be set low. If it is set to high, the search will turn into a primitive random search.

The classic example of a mutation operator involves a probability that an arbitrary bit in a genetic sequence will be changed from its original state. A common method of implementing the mutation operator involves generating a random variable for each bit in a sequence. This random variable tells whether or not a particular bit will be modified. This mutation procedure, based on the biological point mutation, is called single point mutation. Other types are inversion and floating point mutation. When the gene encoding is restrictive as in permutation problems, mutations are swaps, inversions and scrambles.

The purpose of mutation in GAs is preserving and introducing diversity. Mutation should allow the algorithm to avoid local minima by preventing the population of chromosomes from becoming too similar to each other, thus slowing or even stopping evolution. This reasoning also explains the fact that most GA systems avoid only taking the fittest of the population in generating the next but rather a random (or semi-random) selection with a weighting toward those that are fitter.

3. Genetic algorithm applied to phase recovery

The fringe demodulation problem is difficult to solve since many solutions are possible, even for a single noiseless fringe pattern.

The image on a fringe pattern $I(x,y)$ does not change if $\varphi(x,y)$ in Eq.(1) is replaced by another phase function $\hat{\varphi}(x,y)$ given by:

$$\hat{\varphi}(x, y) = \pm \varphi(x, y) \pm 2\pi k, \quad (2)$$

A fringe pattern of $R \times C$ pixels dimension is segmented into a window overlapping set of sub-images of $R1 \times C1$ pixels dimensions, and with origin coordinates at (r, c) . The GA is used to carry out the optimization process, where a parametric estimation of a non linear function is proposed to fit the phase on the sub-images.

The fitness function is modelled by the next 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.

The fitting function is chosen depending on prior knowledge of the demodulation problem, but when no prior information about the shape of $\varphi(x, y)$ is known, a polynomial fitting is adequate. The adequate dimensionality of the polynomial depends on the interferogram complexity, but if we only want to estimate the phase in a small region, the dimensionality of the function can be low.

A r -degree approximation is used so that the phase data can be modelled like:

$$p(\mathbf{a}, x, y) = a_0 + a_1x + a_2y + a_3x^2 + a_4y^2 + a_5xy + a_6x^2y + a_7xy^2 + a_8x^3 + a_9y^3 + \dots + a_qy^r \quad (3)$$

3.1 Decoding chromosomes

As it was said earlier, the GA is used to find the function parameters, in this case, vector \mathbf{a} . If we use this function, the chromosome can be represented by the vector:

$$\mathbf{a} = [a_0 \ a_1 \dots a_q] \quad (4)$$

A k -bit long bit-string is used to codify an allele; then, the chromosome 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 i -th parameter is:

$$a_i = L_i^B + \frac{(L_i^U - L_i^B)}{2^k - 1} N_i, \quad (5)$$

where a_i is the i -th parameter real value, L_i^B is the i -th bottom limit, L_i^U is the i -th upper limit, and N_i is the decimal basis value of the bit-string.

L_i^B and L_i^U are redefined for each sub-image. The maximum value for each parameter is calculated taking into consideration the maximal phase value on that window. These maximum values can be expressed as:

$$L_0^B = -\pi \quad \text{and} \quad L_0^U = \pi \quad (6)$$

$$L_i^U = -L_i^B \quad (7)$$

$$L_i^U = \frac{4\pi F}{R1_i^m C1_i^n} \quad (8)$$

where F is twice the maximum fringe number on the window:

$$F = 2 \cdot \max\left(F_x, F_y, \sqrt{F_x^2 + F_y^2}\right) \quad (9)$$

F_x and F_y are the maximum fringe numbers in the x and y directions. m is the relative grade for x of the i -th term, and n is the relative grade for y of the i -th term.

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

$$\mathbf{a}' = [a_0 \ a_1 \dots a_q] \quad (10)$$

so, $p(\mathbf{a}, x, y)$ can be expressed as:

$$p(\mathbf{a}, x, y) = p(\mathbf{a}', x, y) + a_0 \quad (11)$$

and replacing Eq. 11 into Eq 1:

$$I(x, y) = a(x, y) + b(x, y) \cos(p(\mathbf{a}', x, y) + a_0) \quad (12)$$

Additionally, a_0 can be expressed as $a_0 = 2\pi l + a_0'$, l is an integer, and, $a_0' < 2\pi$ so Eq. 12 becomes:

$$I(x, y) = a(x, y) + b(x, y) \cos(p(\mathbf{a}', x, y) + a_0' + 2\pi l) \quad (13)$$

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

$$I(x, y) = a(x, y) + b(x, y) \cos(p(\mathbf{a}', x, y) + a_0') \quad (14)$$

Eq. 14 demonstrates that limits for a_0 within a range of 2π are enough to represent the phase of the fringe pattern.

In the next section, it will be seen that a_0 can be separated from \mathbf{a} and calculate it independently. Mutation and crossover operators can be applied only over \mathbf{a}' , and then add the calculated value for a_0 to \mathbf{a}' , and recover vector \mathbf{a} .

3.2 Fitness function

The GA, as was described in section 2, is an optimization procedure. Fitness function is always positive, and the optimal point is one minimum value. A negative sign transforms a problem from minimization to maximization. In a given generation, the maximal and minimal values or the fitness values are searched, and they are used to linearly adjust the dynamic range from 0 to A. The values are now positive, and are called aptitude. x and y

are the coordinates in the fringe images. (r, c) are the absolute coordinates of the origin coordinate of the sub-image, f is the function that is adjusted in the current window to approximate the phase term.

The fitness function is applied over each window and swept over the entire image. The path that the process follows can be an arbitrary choice. It is recommended that the window has between 40% and 60% overlapped area with previously demodulated data. This condition is required so the new demodulated phase can be coupled to the previously demodulated phases.

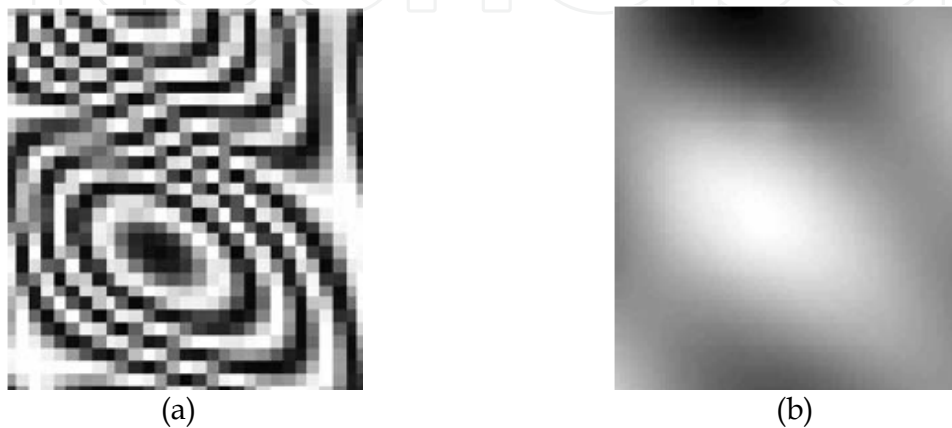


Fig. 9. (a) Fringe pattern subsampled. (b) Demodulated phase.

A fitness function $U(\mathbf{a}^p)$ for each sub-image is used to obtain the fitness value for the p -th chromosome in the population. Its form could be diverse, but most of them have a term that compares the RMS error between the original fringe pattern and the fringe pattern obtained from the estimated phase (similarity):

$$U(\mathbf{a}^p) = \sum_{y=r}^{r+R1} \sum_{x=c}^{c+C1} \left[I_N(x, y) - \cos(f(\mathbf{a}^p, x, y)) \right]^2 \quad (15)$$

Additional terms are added to the fitness functions to give restrictions on the proposed phase. In (Cuevas et al, 2006) fitness function has three criteria: similarity, smoothness and overlapped phase similarity with a previously estimated phase.

$$R(\mathbf{a}^p) = \sum_{y=r}^{r+R1} \sum_{x=c}^{c+C1} \left\{ \begin{aligned} &\lambda_1 \left[\left(f(\mathbf{a}^p, x-r, y-c) - f(\mathbf{a}^p, x-1-r, y-c) \right)^2 \right. \\ &\quad \left. + \left(f(\mathbf{a}^p, x-r, y-c) - f(\mathbf{a}^p, x-r, y-1-c) \right)^2 \right] m(x, y) \\ &\quad \left. + \lambda_2 \left[\left(f(\mathbf{a}^p, x-r, y-c) - \Phi(x, y) \right)^2 \right] \right\} \quad (16) \end{aligned} \right.$$

$R(\mathbf{a}^p)$ is the total amount the restrictions add to the fitness function, for a given window whose origin is on (r, c) ; $m(x, y)$ is a mask that indicates where, inside the image, exist the

fringe pattern, $n(x, y)$ indicates the overlapping zone between the phase of a given window, φ , and the total phase, Φ .

The third term, associated with λ_2 , allow the GA to extrapolate the trend of the adjacent demodulated regions into the actual window. As a consequence, the GA can demodulate interferograms that are noisy and are subsampled.

In WFPD (Cuevas et al, 2006) the fitness function has three criteria: similarity, smoothness and overlapped phase similarity with a previously estimated phase. It is eliminated third criterion, this simplified fitness function can make robust the phase retrieval in one window from demodulation errors in another window. The phase in different windows can be demodulated in parallel. The resulting phase segments are splicing sequentially. Noise filtering and fringe normalization are solved using alternative low-pass filtering techniques. We suppose a smooth phase with continuity in first and second derivatives.

A fitness function $U(\mathbf{a}^p)$ for each sub-image is used to obtain the fitness value for the p -th chromosome in the population, it can be written as:

$$\begin{aligned}
 U(\mathbf{a}^p) = & - \sum_{y=r}^{r+R1} \sum_{x=c}^{c+C1} \left\{ \left[I_N(x, y) - \cos(f(\mathbf{a}^p, x, y)) \right]^2 \right. \\
 & \lambda_1 \left[\left(f(\mathbf{a}^p, x, y) - f(\mathbf{a}^p, x-1, y) \right)^2 \right. \\
 & \left. \left. + \left(f(\mathbf{a}^p, x, y) - f(\mathbf{a}^p, x, y-1) \right)^2 \right] \right. \\
 & \lambda_2 \left[\left(f(\mathbf{a}^p, x+1, y) - f(\mathbf{a}^p, x-1, y) \right)^2 \right. \\
 & \left. \left. + \left(f(\mathbf{a}^p, x, y+1) - f(\mathbf{a}^p, x, y-1) \right)^2 \right] \right\} m(x, y).
 \end{aligned} \tag{17}$$

where $f(\mathbf{a}^p, x, y) = p(\mathbf{a}^p, x, y)$, λ_1 and λ_2 are the regularization terms used to penalize high first and second phase derivatives, and to assure the smoothness of the phase. $m(x, y)$ is a binary mask which defines the valid and invalid data image area. λ_1 and λ_2 values are chosen empirically, and are dependent on window size and fringe pattern regional frequency contents. Typical values are 0.005 to 0.07 for λ_1 and 0.00025 to 0.002 for λ_2 for a 7x7, 9x9 or 11x11 size window.

The following terms were used in the fitness function:

a. Fringe similarity criterion:

$$\left[I_N(x, y) - \cos(f(\mathbf{a}^p, x, y)) \right]^2 \tag{18}$$

The fringe pattern is considered to be normalized on the range [-1,1], or a binary version of the original. There are several methods in the literature for the normalizing of binary threshold fringe patterns [16]. The normalized genetic fringe image is calculated using the cosine of the fitting function. The sum of the squared differences between original fringes

and the synthesized fringe pattern is calculated, and those chromosomes that generated a more similar fringe pattern to the original, will have a higher probability of being selected.

As it was said earlier, parameter a_0 can be calculated so that mutation and crossover are only applied to \mathbf{a}' . a_0 is spliced onto \mathbf{a}' to recover parameter vector \mathbf{a} .

The missing parameter, a_0 , is obtained from the fringe similarity criterion by:

$$\begin{aligned} \frac{\partial U_s}{\partial a_0} &= A \sin(a_0) + B \cos(a_0) - C \cos(2a_0) - D \sin(2a_0) \\ \text{where} \\ A &= \sum 2I_N(x, y) \cos(\mathbf{a}'), \\ B &= \sum 2I_N(x, y) \sin(\mathbf{a}'), \\ C &= \sum \sin(2\mathbf{a}'), \\ \text{and} \\ D &= \sum \cos(2\mathbf{a}'). \end{aligned} \quad (19)$$

A , B , C , and D are constants for any value given to a_0 . a_0 is chosen to be the value that makes $\frac{\partial U_s}{\partial a_0} = 0$ in the domain $[-\pi, +\pi]$. If we define the function:

$$f(a_0) = \left(\frac{\partial U_s}{\partial a_0} \right)^2 \quad (20)$$

the problem is finding the minimum. There are several methods to find it, like the Newton method, the Fibonacci search, the steepest descent method, etc. The search domain in Eq. 20 is well defined and small enough, so an exhaustive search can easily found the desired value of a_0 .

b. Smoothness criterion:

$$\begin{aligned} \lambda_1 &\left[\left(f(\mathbf{a}^p, x, y) - f(\mathbf{a}^p, x-1, y) \right)^2 \right. \\ &\quad \left. + \left(f(\mathbf{a}^p, x, y) - f(\mathbf{a}^p, x, y-1) \right)^2 \right] \\ \lambda_2 &\left[\left(f(\mathbf{a}^p, x+1, y) - f(\mathbf{a}^p, x-1, y) \right)^2 \right. \\ &\quad \left. + \left(f(\mathbf{a}^p, x, y+1) - f(\mathbf{a}^p, x, y-1) \right)^2 \right] \end{aligned} \quad (21)$$

The weighted sum of the discrete approximation (squared differences) of the first and second derivatives is calculated. The goal is to achieve smooth solutions in the first and second derivatives. This term contributes in a negative way to the maximization fitness process, so the chromosome decreases its fitness value.

This term is necessary because of the use of a polynomial function. We use a high degree polynomial interpolation, and oscillations are present on the estimated phase; they can introduce errors into the demodulation process. λ_1 and λ_2 are chosen to minimize these oscillations.

3.3 Selection operator

Chromosomes are evaluated using the fitness function. The result of the evaluation is their fitness value. Fitness value of the entire population is stretched in the range 0 to A. In each generation, the minimum and maximum fitness values are obtained to produce a normalized fitness value:

$$NFv(i)_c = \left(FV(i)_c - \min(i) \right) \frac{A}{\max(i) - \min(i)} , \quad (22)$$

where $NFv(i)_c$ is the normalized fitness value in the i -th generation for the c -th chromosome, and $\min(i)$ and $\max(i)$ are the minimum and maximum values of $FV(i)_c$, the fitness value for a member in the i -th generation.

The normalized fitness value is used to calculate the probability selection P_s of a given chromosome. In order to represent a sexual reproduction, pairs of chromosomes are used to produce a new population. Two chromosomes are randomly selected with a P_s that is proportional to its normalized fitness value.

For the c -th chromosome, a random number r_s is generated. If $r_s < P_s$, the chromosome is selected. Two chromosomes are picked this way, and the crossover operator is applied over them to create two new chromosomes for the new population.

P_s can be calculated in many ways. The easiest is the Roulette Wheel method, which P_s is calculated by:

$$P(i)_s = \frac{NFv(i)_s}{\sum_s NFv(i)_s} \quad (23)$$

The Roulette wheel method has the disadvantage of being able to produce a premature convergence, so a Boltzmann selection method was used to avoid this inconvenient:

$$P(i)_s = \frac{\exp\left[\left[NFv(i)_s\right]/T(i)\right]}{\sum_s \exp\left[\left[NFv(i)_s\right]/T(i)\right]} , \quad (24)$$

where $T(i)$ $T(i)$ is the temperature in the i -th generation. $T(0)$ is a large value, so P_s is similar for all chromosomes in the initial generation. T is decreased over the generations, so the P_s for the best adapted is progressively higher than for the least adapted. In this way, there is the opportunity to explore all the space of solution, so the probability of falling into a local minimum is lowered. T is varied by:

$$T(i) = T_0 \exp(-i / k) \quad (25)$$

where k is a constant that indicates in which generation $T(i) \approx T_0 / 3$. The process of selection and crossover is repeated until an entire new population is obtained.

3.4 Crossover operator

In the GA, a Crossover probability P_c is given to exchange the genetic information between two chromosomes, so that if a randomly generated number is smaller than it, the chromosomes are mixed to produce two new individuals, and if the number is bigger than P_c , the two original chromosomes are added to the new population. In the phase recovery from a fringe pattern, a two point crossover was used, where two crossover points are randomly generated. In this case, it is required to swap the central segments between chromosomes.

3.5 Mutation operator

Mutation is the best known mechanism to produce variations. Alleles of the chromosomes are randomly replaced by others in a random way. Mutation is treated like a background operator to ensure variety in the population.

In GA, a mutation probability P_m is defined. For each position, a random number is generated, and if it is smaller than P_m , the allele is changed for another. In a binary chromosome code, a '0' is changed for '1' and a '1' is changed for '0'.

3.6 GA convergence

GA convergence depends mainly on population size. With a large population, convergence is achieved in a smaller number of iterations, but the processing time is increased. To stop the GA process, different convergence measures can be employed. The maximum number of iterations is chosen at B . The algorithm can be stopped when a relative error ε is smaller than a predefined limit δ :

$$\varepsilon = \left| \frac{\max(i) - \max(i-1)}{\max(i)} \right|. \quad (26)$$

3.7 Splicing process

As it was mentioned before, phase demodulation is achieved through window segmentation of the fringe pattern. The GA demodulates the phase inside each window, independently from others, so this process can be made in parallel. It is supposed that the demodulated phase field in each window is differentiated from the corresponding real phase field only by the concavity and the DC bias. Then, a splicing procedure is required to connect different GA fitted phase windows and determine the whole phase field $\Phi(x, y)$. The splicing process is carried out in a sequential way (e.g., row by row).

It is described as follows:

1. The demodulated phase from the first window is used as the initial reference.

- From the GA current fitted phase window $\varphi(x,y) = f(\mathbf{a}^p, x, y)$, a second phase field is calculated $\varphi'(x,y)$ with a negative concavity) as $\varphi'(x,y) = -\varphi(x,y)$, or $\varphi'(x,y) = f(-\mathbf{a}^p, x, y)$.
- Two DC bias are calculated, one for $\varphi(x,y)$ and one for $\varphi'(x,y)$ using:

$$DC_1 = \frac{\sum_{x,y \in N} (\Phi(x,y) - \varphi(\mathbf{a}^p, x, y))}{A} \text{ and } DC_2 = \frac{\sum_{x,y \in N} (\Phi(x,y) - \varphi'(\mathbf{a}^p, x, y))}{A}, \quad (27)$$

where N is the overlapped neighbourhood region, and A is the overlapped area (pixel²) of N .

- The RMS error for the two alternative phase window fields, $\varphi(x,y)$ and $\varphi'(x,y)$, compared against $\Phi(x,y)$ is calculated as:

$$RMS_1 = \frac{\sum_{x,y \in N} (\Phi(x,y) - \varphi(\mathbf{a}^p, x, y) - DC_1)^2}{A} \text{ and } RMS_2 = \frac{\sum_{x,y \in N} (\Phi(x,y) - \varphi'(\mathbf{a}^p, x, y) - DC_2)^2}{A} \quad (28)$$

- The phase described by the function with the minimum RMS error value ($\varphi + DC_1$ or $\varphi' + DC_2$) is spliced onto the demodulated phase field Φ .
- If there are more windows to splice, the next window in the sequence is labelled as the current window and goes to step 2. Otherwise, the splicing process is finished.

4. Adjustable genetic algorithm

In previous works, the window has a fixed dimension. The RMS error for a given window varies due to frequency content of the image, the window size and the values given in the smoothness criterion. High frequency zones are best demodulated using small windows; the demodulating process in low frequency zones is better when using a large window. In fact, using a small window in low frequency zones introduces unnecessary noise due to the splicing process for a certain region. This frequency estimator is used again to determine the best demodulation path; beginning in low frequency regions, it develops across increasing complexity regions. As a result, regions that are easily demodulated can be used to guide the splicing of higher difficulty regions.

Finally, a modification on the GA is presented to determine the smoothness criterion automatically. In a previous work, some values to λ_1 and λ_2 were proposed to be used for certain window values. In this work, this smoothness criterion is calculated by the GA itself, adding two genes to the chromosome; this codifies this criterion in real numbers. As a result, the GA calculates the coefficient vector of the polynomial and the smoothness parameters needed for the correct estimation of the phase at the same time. This modification allows one to have an algorithm that depends less on external characterization.

4.1 Local frequency estimator

In this section, a method to classify regions on the image according to their frequency content is described. This process is necessary to determine the size of the window needed to demodulate a certain region. A given fringe pattern is segmented into two, three or more regions.

1. From the original image, we obtain the image that shows the gradient. This image is obtained by correlating image $I(x,y)$ of $R \times C$ dimensions with the Sobel operators.

$$S_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} \quad (29)$$

$$S_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (30)$$

The correlation is defined as:

$$C_k(x,y) = \sum_i \sum_j I(i,j) S_k(x+i, y+j) \quad (31)$$

At each point, the magnitude of the gradient is obtained:

$$|S| = \sqrt{S_x^2 + S_y^2} \quad (32)$$

2. A median filter is used to detect the zones where fringe frequency increases. A median filter of dimensions $N \times N$ is defined as:

$$M(x,y) = \frac{\sum_{i=-N/2}^{N/2} \sum_{j=-N/2}^{N/2} I(x+i, y+j)}{N \times N} \quad (33)$$

3. The image obtained by means of this process shows brighter zones in the regions with high frequency. This image is segmented into three regions of low, medium and high frequency content.

The brightest point, $\max(M)$, and the darkest point, $\min(M)$, are located. Their values are used to perform the segmentation of image $M(x,y)$ using:

$$F(x,y) = \frac{[M(x,y) - \min(M)]}{\max(M) - \min(M)} \quad (34)$$

$F(x,y)$ is thresholded in three zones:

$$FT(x,y) = \begin{cases} 0 & 0.3 > \frac{[M(x,y) - \min(M)]}{\max(M) - \min(M)} \geq 0 \\ 1 & 0.7 > \frac{[M(x,y) - \min(M)]}{\max(M) - \min(M)} \geq 0.3 \\ 2 & 1 \geq \frac{[M(x,y) - \min(M)]}{\max(M) - \min(M)} \geq 0.7 \end{cases} \quad (35)$$

Each zone indicates the relative level of frequency content in $I(x,y)$. $F(x,y)$ and $FT(x,y)$ are used to obtain the demodulation map, and to indicate the window size suited for a certain region.

4.2 Quality map and demodulation path

The windows where the polynomial is adjusted are extracted from $I(x,y)$ following the fringes, like in the phase tracker method. To achieve this result, in a previous work a quality map is obtained from $I(x,y)$ by thresholding it on Q levels, and using a filling algorithm to obtain a continuous path along the fringes.

In this work, we propose a method to obtain a different quality map. It is desirable that the demodulation path fulfils more conditions other than following the fringes. The conditions proposed are:

- Follow the fringes.
- Follow the frequency contents from low frequencies to high frequencies.
- Sequentially demodulate regions with increasing levels of difficulty.

We postulate that criterions b and c are identical, so the problem is reduce to incorporating the second criterion into the quality map.

To achieve this goal, $F(x,y)$ is used. $F(x,y)$ is added to $I(x,y)$, and the new image $IF(x,y)$ is used to obtain the quality map, by thresholding it on Q levels.

$$IF(x,y) = I(x,y) + F(x,y) \quad (36)$$

$$IFT(x,y) = Q \frac{[IF(x,y) - \min(IF)]}{\max(IF) - \min(IF)} \quad (37)$$

IFT is used to generate a new demodulation path that begins on low frequencies and follows the fringes until it reaches high frequencies.

The algorithm used to generate the demodulation path is :

- An array with all point labelled with some level (quality map $IFT(x,y)$) is needed
- In the frequency map, $F(x,y)$, the minimum frequency area is searched, and the point that is $\min(F)$ is chosen.
- A matrix $L(x,y)$ of $R \times R$ dimensions is defined. All its values are set to '0'.
- An array of stacks $[S_1, S_2, \dots, S_Q]$ is defined. All stacks have $R \times C$ dimensions. The initial point is placed on the stack corresponding to its value. i.e. if $IFT(x,y) = 1$, then coordinates (x,y) are stored in stack S_1 .
- An array of coordinates $C(n)$ of $R \times C$ dimensions is defined. The coordinates of the first point are stored in it. An index n is defined, and its value is set to '0'. $L(x,y)$ is set to '1':

$$\begin{aligned} n &\leftarrow 0 \\ C(n) &\leftarrow (x,y) \\ L(x,y) &\leftarrow 1 \end{aligned} \quad (38)$$

- f. The points surrounding (x, y) are scanned while varying index i and index j from -1 to +1, and their coordinates stored in their corresponding stacks:

$$\begin{aligned} &\text{if } L(x+1, y+1) = 0 \\ &S_{IFT(x+i, y+j)} \leftarrow (x+i, y+j) \end{aligned} \quad (39)$$

- g. Beginning on S_1 , a non-empty stack is searched. The point (x, y) on top of that stack S_i is brought out. The next actions are performed:

$$\begin{aligned} n &\leftarrow n+1 \\ C(n) &\leftarrow (x, y) \\ L(x, y) &\leftarrow 1 \end{aligned} \quad (40)$$

- h. If $n \leq R \times C$ steps 'f' and 'g' are repeated. Points stored in $C(n)$ indicate the demodulation path to follow.

4.3 Genetic algorithms internal parameters automatic adjust

GAs have great possibilities to develop robust algorithms, varying their internal parameters according to the task to be performed. In this paper, we propose using GAs to adjust the internal parameters of this algorithm.

4.3.1 Regularization terms

The original image is shown in fig. 10. A media filter is applied onto the gradient image, yielding high values in regions with high fringe density.

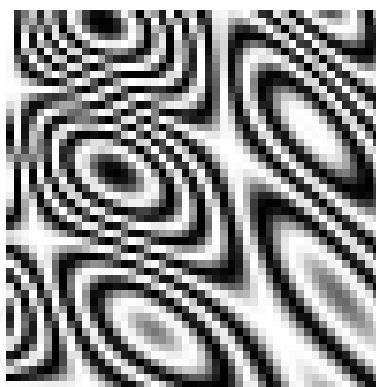


Fig. 10. Original image.

Maximum and minimum values are localized, and the image is discretized into three main regions. A given window size is assigned to each region (Fig 11).

When the demodulation window is positioned in a given region, the size window is varied to obtain a better adjustment. The last stage is to introduce these regions into the demodulation process.

A quality map is used to determine the demodulation path. Low frequency regions are demodulated first by adding the discretized map of frequencies to the original image. A filling algorithm is used to determine the demodulation path using this quality map.

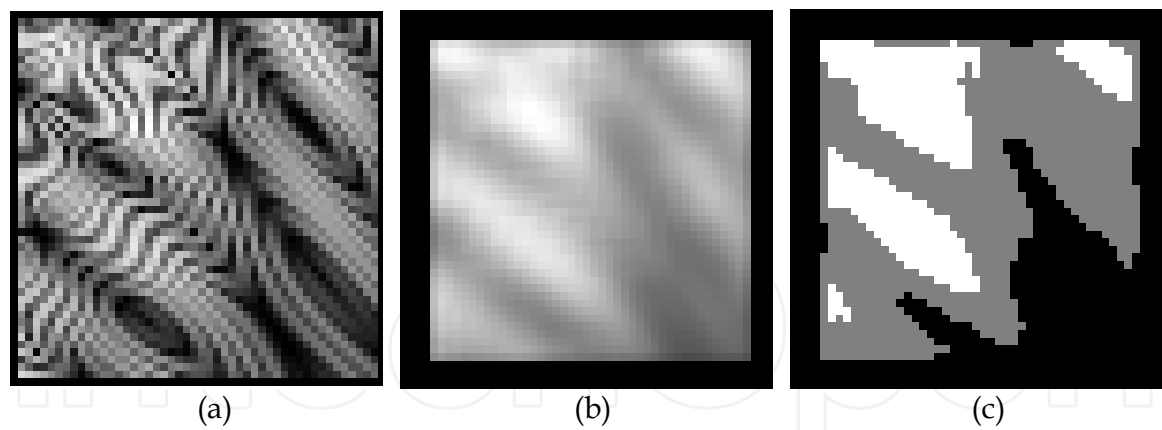


Fig. 11. Gradient (a) smoothed with a 9x9 window (media filter) (b) Classification on high, medium and low frequencies.

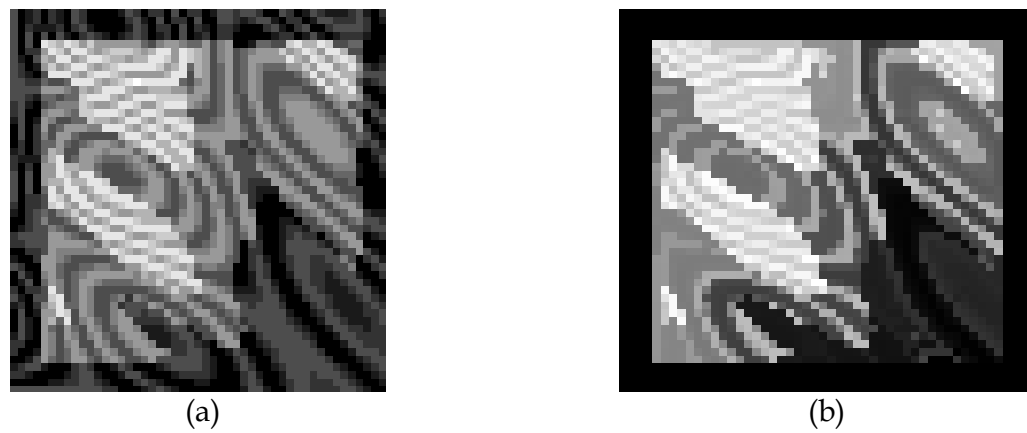


Fig. 12. (a) Quality map, 15 levels; (b) Demodulation path that follows fringes and increased frequency regions.

5. Results

The proposed algorithm is used to demodulate a shadow moiré image taken from a modeled figure of a dolphin. The noise in the original shadow moiré is filtered through the use of a median filter. This filter is used only on an area determined by a mask that indicates the allowed area on which to perform the demodulation process.

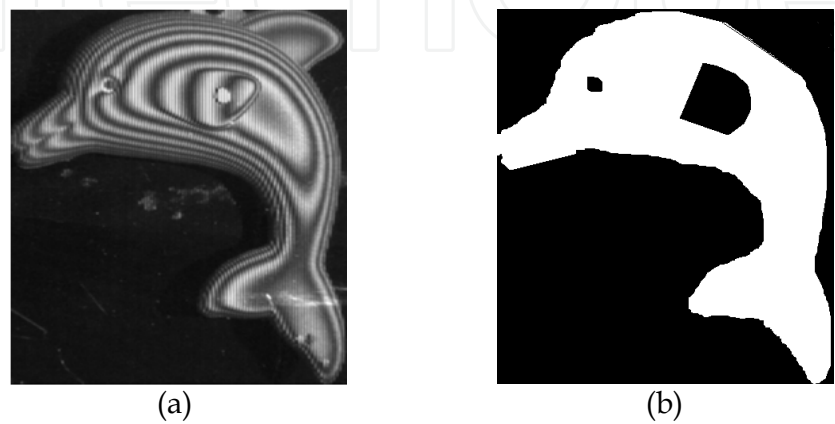


Fig. 13. (a) Fringe pattern shadow moiré; (b) Mask $m(x,y)$.

The algorithm needs surfaces that are continuous on the first and second derivatives, so, regions in the original dolphin figure that do not fulfill these requirements are cut from the mask.



Fig. 14. Image after noise filtering using a median filter.



Fig. 15. (a) Result of binarize fig 14; (b) Fringe pattern associated with the demodulated phase.

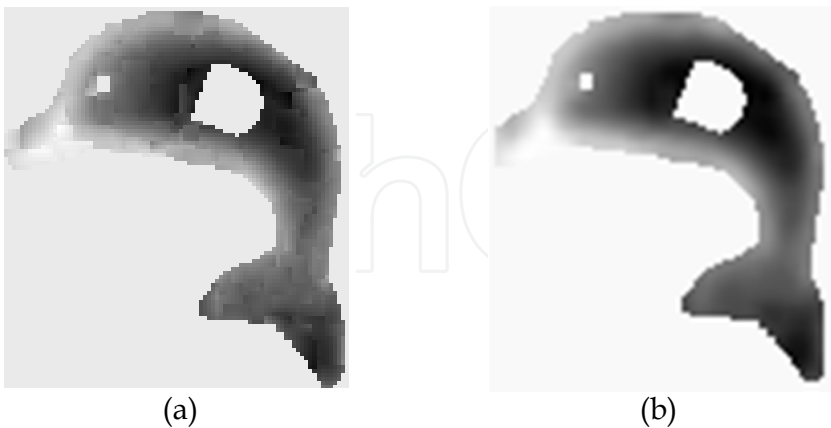


Fig. 16. (a) Phase demodulated from fig. 15(a); (b) Phase smoothed using a media filter 7x7.

The resulting image is binarized to reduce noise interference in the demodulation process. This binarized image is fed to the algorithm.

The demodulated phase is smoothed using a 7x7 media filter, after using a contrast enhancement and a low pass filter.

6. Conclusions

Computational power has been increased in the last years. This increased processing power allows to develop EA as a practical tool with application in pattern recognition, computational intelligence, image processing, automatization, and others. EA's, fuzzy logic and neural networks are called soft computing, because they can deal with problems where there are not a good knowledge of the problem, information is incomplete or inconsistent, or are large amount of noise.

In this chapter is shown only a single application of EA on fringe pattern demodulation. There still a lot of variations that can be explored to improve the performance of actual algorithms.

GA based methods have two advantages over regularized phase tracker: they can work on low resolution images and they can follow changes in concavity. These advantages are the consequence of taking upper grade terms in the interpolated function.

The technique showed in (Toledo, Cuevas 2009), called FPIW, is based in two suppositions: it is not necessary to know the phase on the neighborhood to estimate the phase in a given window, and the estimated phase in a window differ only by its concavity sign and a DC bias, from the real phase in the region framed by the window. As a consequence, the overlapped similarity criterion used in the WFPD (Cuevas et al, 2003) method can be eliminated from the fitness function in the FPIW method. In exchange, FPIW works near Nyquist, but on sub-Nyquist, WFPD is better.

The phase in a given window is estimated without known nothing about the phase in other windows. It is possible to demodulate simultaneously all windows, that is, FPIW method described has implicit parallelism. WFPD demodulate the windows sequentially.

7. Acknowledgments

We acknowledge the support of "Consejo Nacional de Ciencia y Tecnologia" of Mexico, "Consejo de Ciencia y Tecnología del Estado de Guanajuato," and "Centro de Investigaciones en Optica, A.C." We also thanks Mario Ruiz Berganza for his aid proofreading this paper; and thanks to Guillermo Garnica for his invaluable technical support. H. Sossa thanks "Consejo Nacional de Ciencia y Tecnologia" for grant 155014 and "SIP-IPN" for grant 20121311.

8. References

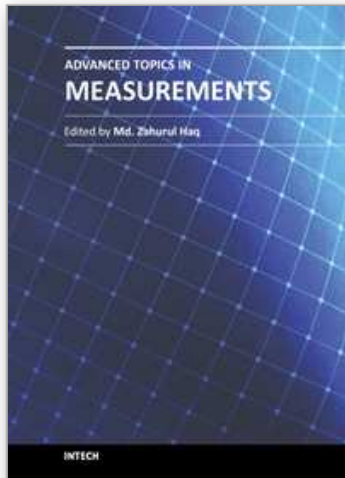
- Bäck T, Fogel DB, Michalewicz Z (2000) *Evolutionary computation*. Institute of Physics Publishing, Bristol.
- Bone, D.J., *Fourier fringe analysis: the two dimensional phase unwrapping problem*, Appl. Opt, 30, 3627-3632, 1991,
- Buckland, J.R., Huntley, J.M., and Turner, S.R.E., *Unwrapping noisy phase maps by use of a minimum-cost-matching algorithm*, Appl. Opt., Vol. 34, No. 23 (August 1995) pp 5100-8.

- Cuevas FJ, Servin M, Rodríguez-Vera R (1999) *Depth recovery using radial basis functions*. Opt Communication 163:270–277.
- Cuevas FJ, Servin M, Stavroudis ON, Rodríguez-Vera R (2000). *Multi layer neural network applied to phase and depth recovery from fringe patterns*. Opt Commun 181:239–259.
- Cuevas F.J., Mendoza F., Servin M., Sossa-Azuela J.H. (2006) *Window fringe pattern demodulation by multi-functional fitting using a genetic algorithm*. Opt. Commun. 261:231–239.
- Cuevas FJ, Sossa-Azuela JH, Servin M (2002) *A parametric method applied to phase recovery from a fringe pattern based on a genetic algorithm*. Opt Commun 203:213–223.
- Fernández A, Kaufmann GH, Doval AF, Blanco-García J, Fernández JL (1998) *Comparison of carrier removal methods in the analysis of TV holography fringes by the Fourier transform method*. Opt Eng 37:2899–2905.
- Ghiglia, D.C., and Romero, L.A., *Robust two dimensional weighted and unweighted phase unwrapping that uses fast transforms and iterative methods*, J. Opt. Soc. Am. A, 11, 107–117, 1994.
- Goldberg D (1989) *Genetic algorithms: search and optimization algorithms*. Addison-Wesley, Reading, MA.
- Holland JH (1975) *Adaptation in natural and artificial systems*. University of Michigan Press, Michigan.
- Ichioika Y, Inuiya M (1972) *Direct phase detecting system*. Appl Opt 11:1507–1514.
- Juan Antonio Quiroga, José Antonio Gómez-Pedrero, Ángel García-Botella, *Algorithm for fringe pattern normalization*, Optics Communications, Volume 197, Issues 1-3, 15 September 2001, Pages 43-51
- Juan Antonio Quiroga, Manuel Servin, *Isotropic n-dimensional fringe pattern normalization*, Optics Communications, Volume 224, Issues 4-6, 1 September 2003, Pages 221-227.
- L.E. Toledo & F.J. Cuevas, *Optical Metrology by Fringe Processing on Independent Windows Using a Genetic Algorithm*, Springer Verlag Experimental Mechanics 48, pp 559-569.
- Larkin KG, Bone DJ, Oldfield MA (2001) *Natural demodulation of two-dimensional fringe patterns in general background of the spiral phase quadrature transform*. J Opt Soc Am A 18:1862–1870.
- Malacara D, Servin M, Malacara Z (1998) *Interferogram analysis for optical testing*. Marcel Dekker, New York.
- Noé Alcalá Ochoa, A.A. Silva-Moreno, *Normalization and noise-reduction algorithm for fringe patterns*, Optics Communications 270: 161-168.
- Quiroga JA, Gómez-Pedrero JA, García-Botella A (2001) *Algorithm for fringe pattern normalization*. Opt Communications 197:43.
- Servin M, Rodríguez-Vera R (1993) *Two dimensional phaselocked loop demodulation of interferogram*. J Mod Opt 40:2087–2094.
- Servin M, Marroquín JL, Cuevas FJ (2001) *Fringe-follower regularized phase tracker for demodulation of closed-fringe interferograms*. J Opt Soc Am A 18:689–695.
- Servin M, Quiroga JA, Cuevas FJ (2001) *Demodulation of carrier fringe pattern by the use of non-recursive digital phase locked loop*. Opt Commun 200:87–97.

Takeda M, Ina H, Kobayashi S (1982) *Fourier-transform method of fringe-pattern analysis for computer based topography and interferometry*. J Opt Soc Am 72:156.

IntechOpen

IntechOpen



Advanced Topics in Measurements

Edited by Prof. Zahurul Haq

ISBN 978-953-51-0128-4

Hard cover, 400 pages

Publisher InTech

Published online 07, March, 2012

Published in print edition March, 2012

Measurement is a multidisciplinary experimental science. Measurement systems synergistically blend science, engineering and statistical methods to provide fundamental data for research, design and development, control of processes and operations, and facilitate safe and economic performance of systems. In recent years, measuring techniques have expanded rapidly and gained maturity, through extensive research activities and hardware advancements. With individual chapters authored by eminent professionals in their respective topics, Advanced Topics in Measurements attempts to provide a comprehensive presentation and in-depth guidance on some of the key applied and advanced topics in measurements for scientists, engineers and educators.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

L. E. Toledo, F. J. Cuevas, J.F. Jimenez Vielma and J. H. Sossa (2012). Fringe Pattern Demodulation Using Evolutionary Algorithms, Advanced Topics in Measurements, Prof. Zahurul Haq (Ed.), ISBN: 978-953-51-0128-4, InTech, Available from: <http://www.intechopen.com/books/advanced-topics-in-measurements/fringe-pattern-demodulation-using-genetic-algorithms>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2012 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the [Creative Commons Attribution 3.0 License](https://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IntechOpen

IntechOpen