Article **Journal of Computational Systems and ICTs**
December 2019 Vol.5 No.16 1-8

Seven frame phase-shifting algorithms for the three-dimensional profilometry using fringe projection and soft computing

Algoritmos de desplazamiento de fase de siete cuadros para la perfilometría tridimensional usando proyección de franjas y cómputo suave

CRUZ-LANDEROS, Jose Juan[1]†*, MOSIÑO, Juan Francisco[1], MARTÍNEZ-REBOLLAR, Alicia[2], BALTAZAR-FLORES, María del Rosario[1]

[1]*Instituto Tecnológico de León*
[2]*National Center for Research and Technological Development*

ID 1st Author: *Jose Juan, Cruz-Landeros* / **ORC ID:** 0000-0002-9110-2007, **CVU CONACYT ID:** 859653

ID 1st Coauthor: *Juan Francisco, Mosiño* / **ORC ID:** 0000-0002-3925-3051, **CVU CONACYT ID:** 25513

ID 2nd Coauthor: *Alicia, Martínez-Rebollar* / **ORC ID:** 0000-0002-1071-8599, **CVU CONACYT ID:** 217272

ID 3st Coauthor: *María del Rosario, Baltazar-Flores* / **ORC ID:** 0000-0002-8847-8732

DOI: 10.35429/JCSI.2019.16.5.1.8 Received July 23, 2019; Accepted September 18, 2019

Abstract

In this paper, we present a set of seven-frame algorithms for the phase shifthing profilometry and their application for the three-dimensional surface imaging using soft computing. This set obtained generate the most common seven-frame algorithms in the state of art. The algorithms are analized for the compensation of common errors such as phase shifthing, phase errors because variation of light or bias.

Fringe analysis, Phase-shifting interferometry, Softcomputing

Resumen

En este artículo se presenta una familia de algoritmos de siete cuadros para la perfilometria de desplazamiento de fase y su aplicación para la obtención de perfiles tridimensionales usando cómputo suave. Esta familia obtenida genera la mayoría de los algoritmos de siete cuadros más comunes en el estado del arte. Los algoritmos son analizados en la compensación de errores comunes como el desentonamiento de fase, errores de fase debido a la variación de iluminación o bias.

Análisis de franjas, Interferometría de corrimiento de fase, Cómputo suave

Citation: CRUZ-LANDEROS, Jose Juan, MOSIÑO, Juan Francisco, MARTÍNEZ-REBOLLAR, Alicia, BALTAZAR-FLORES, María del Rosario. Seven frame phase-shifting algorithms for the three-dimensional profilometry using fringe projection and soft computing. Journal of Computational Systems and ICTs. 2019, 5-16: 1-8.

* Correspondence to Author (email: cruzjj30@gmail.com)
† Researcher contributing first author.

Introduction

The world around us is three-dimensional; however, cameras and image sensors have the ability to acquire two-dimensional representations without depth information. But in recent decades there have been significant advances regarding research, development and commercialization of technologies capable of capturing the depth of a surface. There are numerous applications of these technologies, some of these are facial recognition, creation of exact dental models, simulation of plastic surgeries, reverse product engineering, accident scene investigation (Geng, 2011) and digital fossil conservation (Gutiérrez García, and others, 2015).

The creation of three-dimensional images refers to techniques which are capable of acquiring some properties of three-dimensional objects. Nevertheless, surface acquisition techniques allow obtaining the measurement of the coordinates (x, y, z) of the points on the surface of an object. The result of these techniques can be considered as a map of the depth *z* as a function of the position (x, y). (Geng, 2011)

One of the methods for obtaining surfaces is based on the use of structured light, which is the active lighting of a scene with a special two-dimensional design, varying the intensity patterns. This technique extracts the depth of the object in the scene based on the distortion information of the projected structured light (Geng, 2011), usually captured using a camera and represented in an image (I). A representation of this technique can be seen in Figure 1.

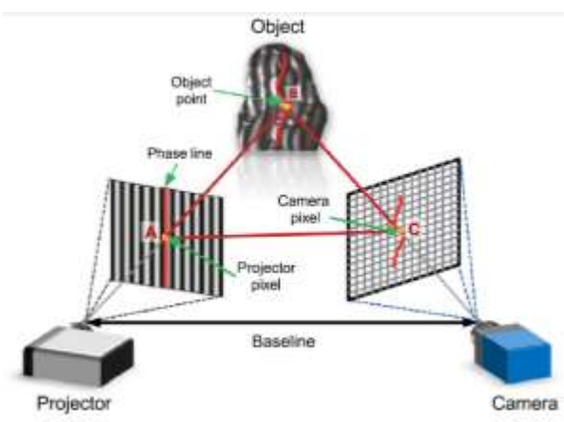


Figure 1 Representation of the structured light projection technique
Source: (Zuo, Huang, Zhang, Chen, & Asundi, 2016)

Phase shifting interferometry is one of the best known structured light projection techniques (Geng, 2011). In one of the stages of this technique, a phase shifting algorithm (PSA) (Gutiérrez García, et al., 2013), also known as a filter, is applied. The objective of this paper is to present the proposal to define filters using a soft computation method known as differential evolution, which is an algorithm inspired by nature used to find values that define a treatable solution to a problem (Simon, 2013). The filters of the experiments performed are defined for scenarios where there is no noise, simulated noise of light variation, vibration in the environment; and these filters are compared with the most used for 7 images.

Phase shifting interferometry

In this technique, a set of *M* sinusoidal patterns is projected on the surface of the object, obtaining a set of *M* images (Geng, 2011). This technique consists of the following steps (Ordoñes Nogales, 2016).

a) *M* images are captured with a constant shifting angle.

b) A phase shifting algorithm of *M* frames is chosen or designed to process the set of images and obtain the wrapped phase.

c) An unfolding algorithm is used or designed to recover the desired phase (the surface).

d) The phase is interpreted in physical measurements.

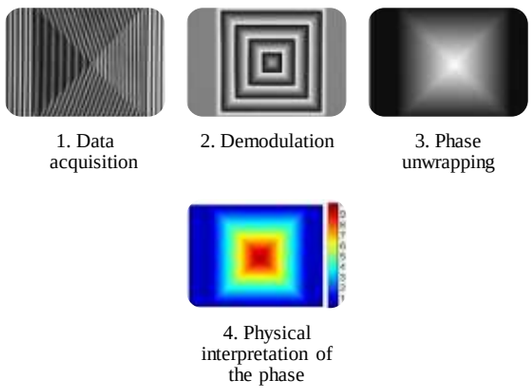


Figure 2 Stages of the phase shifting technique
Source: (Ordoñes Nogales, 2016)

A representation of these steps is seen in Figure 2.

ISSN 2444-5002

In this technique, each image is captured from the surface with the projection of a shifted sinusoidal pattern with a phase angle or $\omega_0 t$ carrier, obtaining a set of M captured images that is represented in an image vector (Equation 1):

$$\boldsymbol{I} = [\boldsymbol{I_1}, \boldsymbol{I_2}, \dots, \boldsymbol{I_M}] \quad (1)$$

In Figure 3, 4 shifted projection patterns are seen at a $\omega_0 t$ angle.

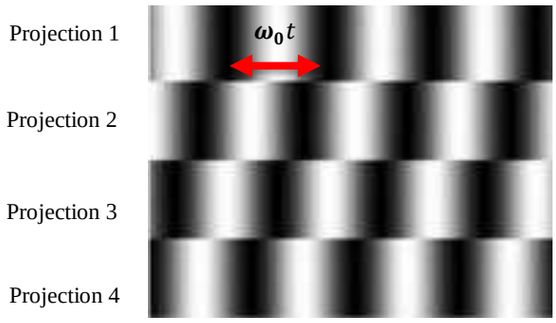


Figure 3 4 shifted projection patterns are seen at a $\boldsymbol{\omega_0 t}$ angle
Source: Prepared by the authors

Then, assuming a small noise value, the set of images obtained can be represented as:

$$\boldsymbol{I}(x, y, t) = a(x, y) + b(x, y)\cos[\phi(x, y) + \omega_0 t] \quad (2)$$

In Equation 2, a is the backlighting, b is the contrast of the patterns (amplitude), $\phi(x, y)$ represents the phase (the z-value of pixel depth (x, y)) and ω_0 is the carrier.

Once the images have been taken, a filter is applied to the M captured images. There are numerous filters that are used in the temporal phase shift (Mosiño, Malacara Doblado, & Malacara Hernández, 2009) (Zuo, Huang, Zhang, Chen, & Asundi, 2016). But filters are designed for uniform spacing and ideal cases; however, when they are applied to real problems, they are not uniform due to the vibration effectors or the variation in the intensity of the projected light. Vibration and light effects are reflected in the values a, b and ω_0 captured by the images.

The tangent of the estimated phase ϕ of any quadrature filter with an M^{th} order (for M images) is given by:

$$\boldsymbol{\tan(\phi)} = \frac{\sum_{k=1}^{M} \boldsymbol{b_k I_k}}{\sum_{k=1}^{M} \boldsymbol{a_k I_k}} = \frac{[\boldsymbol{b_1}\ \boldsymbol{b_2}\ \cdots \boldsymbol{b_M}]\boldsymbol{I}}{[\boldsymbol{a_1}\ \boldsymbol{a_2}\ \cdots \boldsymbol{a_M}]\boldsymbol{I}} = \frac{\boldsymbol{NI}}{\boldsymbol{D\vec{I}}} \quad (3)$$

Where $\boldsymbol{N}$ and $\boldsymbol{D}$ are vectors of coefficients for the numerator and denominator of the filter, respectively. These values define the filter (Mosiño, Malacara Doblado, & Malacara Hernández, 2009). Since the value ϕ is contained in the tan function and it is discontinuous, then a phase development algorithm corresponding to the third stage of this technique is used; this stage is beyond the scope of this work, and since we worked with simulated surfaces, we avoid the need for this technique in experiments.

In (Mosiño, et al., 2011) an expression (Equation 4) was presented to obtain the $\boldsymbol{N}$ and $\boldsymbol{D}$ coefficients that define the filter by establishing the points at which the filter cuts the Fourier space. Where each set $[\alpha_1, \alpha_2, \dots, \alpha_{M-1}]$ defines the frequency to be cut from the Fourier impulse response of the filter. The full explanation of Equation 4 is outside the scope of this work, but can be found in (Mosiño, et al., 2011).

$$\frac{\boldsymbol{N}}{\boldsymbol{D}} = \frac{\sum_{k=1}^{M}\{(-1)^{k+1} \sum \mathbf{sin}[\sum\{\boldsymbol{\alpha_1}, \boldsymbol{\alpha_1}, \dots, \boldsymbol{\alpha_{M-1}}\}_{M-k}^{M-1}]\}}{\sum_{k=1}^{M}\{(-1)^{k+1} \sum \mathbf{cos}[\sum\{\boldsymbol{\alpha_1}, \boldsymbol{\alpha_1}, \dots, \boldsymbol{\alpha_{M-1}}\}_{M-k}^{M-1}]\}} \quad (4)$$

Equation 1

Then, the creation of a filter corresponds to the choice of $[\alpha_1, \alpha_2, \dots, \alpha_{M-1}]$. In this work, the expression of Equation 4 is used to obtain the α_k values that, under certain noise scenarios, better recover the original surface.

Differential evolution

It is one of the most successful methods for continuous optimization (Simon, 2013). It is classified as a soft computing technique the method of which has been inspired by the observation of nature (Simon, 2013); and specifically to the algorithms inspired by the process and mechanism of biological evolution. This process of evolution describes how by means of natural selection (descendants with modifications) a population of a species adapts through generations to certain problems (Brownlee, 2011).

Differential evolution involves the maintenance of a population of candidate solutions for recombination, evaluation and selection.

ISSN 2444-5002

Recombination involves the creation of new candidate components of the solution based on the weighted difference between two randomly selected members of the population and then a third member of the population. The basic differential evolution algorithm is presented in Figure 4.

Input: Parameters: F(scaling factor), CR (crossing constant).
Initialize the population (uniform random distribution);
Repeat
For $(i = 1, i \leq k, i++)$ **Do** /*Each individual*/
Mutate and recombine
$j_{rand} = int(rand_i[0,1] \cdot D) + 1;$
For $(j = 1, j \leq D, j++)$ **Do**
If $(rand_j[0,1] < CR)$ or $(j = j_{rand})$ **Then**
$u_i j = \boldsymbol{v_i j} = \boldsymbol{x_{r3j}} + \boldsymbol{F}(\boldsymbol{x_{r1j}} - \boldsymbol{x_{r2j}})$
Else
$u_{ij} = x_{ij}$
Replace:
$$x_i(t+1) = \begin{cases} u_i(t+1) & \boldsymbol{Si}\ f(u_i(t+1)) \leq f(x_i(t)) \\ x_i(t) & \boldsymbol{Else} \end{cases}$$
End for
Until the termination condition /*example: number of generations*/
Output: Better population or solution.

Figure 4 Pseudocode of the differential evolution algorithm

In this paper, a population of size 40 was used, where each individual corresponds to a vector $\alpha = [\alpha_1, \alpha_2, \alpha_3, \alpha_4]$ that contains the Fourier cut-off points of Equation 4 (for these 7-image experiments, α_6 and α_7 were kept constant to maintain the cyclic filter characteristic in Fourier). The values of α remained in the evolution between 0 and 2π. While the F and CR values observed in the pseudocode were set at 0.8 and 0.9 respectively, these values being recommended in (Simon, 2013), 1000 periods were used in the experimentation.

Methodology

This section describes the experiments performed in this work. They are presented in 4 distinct sections: Both the algorithms and the simulation environment were carried out in IDE Matlab in its R2017b (9.3.0.713579) version.

1.1. Definition of the simulation

The simulation was carried out using the Matlab peaks function (Figure 5) as a surface to be recovered, defined as a 200x200 surface and the reduced height values to 0.15% of the original values of the function to avoid the phase unfolding step in the experiments.

Then, the simulation of the images was done by establishing each one according to the model of the Equation 2, where $\phi(x, y)$ are the peaks surface height values, $a = 0$ and $b = 1$ and where $t = \{-3, -2 - 1, 0, 1, 2, 3\}$.

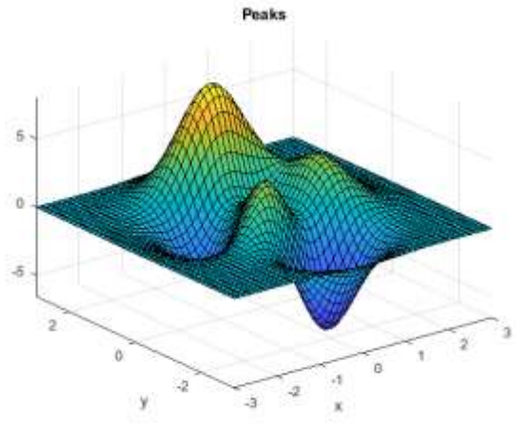


Figure 5 Peaks function in Matlab
Source: Prepared by the authors

1.2. Implementation of the differential evolution algorithm

The differential evolution algorithm shown in Figure 4 was coded. In this implementation, the individuals to be optimized correspond to 4 Fourier cuts of the filter. The filter is generated using Equation 4 and the coefficients that define the filter are obtained to test the profile obtained with the real image. The function to be minimized (Equation 4) is the mean square error between the surface obtained by the filter and the original profile.

$$\boldsymbol{f(\phi_{obtained})} = \left(\boldsymbol{\phi_{obtained}} - \boldsymbol{\phi_{original}}\right)^2 \quad (5)$$

Equation 2

1.3. Evaluation of the coefficients obtained from the filters

The defined coefficients $\boldsymbol{N}$ and $\boldsymbol{D}$ were then used by differential evolution to test the filter in a simulation of the peaks function using 11 different carriers from $\frac{\pi}{10}$ to $\pi - \frac{pi}{10}$ and the quadratic error was obtained between the phase generated by the obtained filter and the original profile.

1.4. Comparison of results with state of the art filters

In this section, 7 new simulations of captured images of the peaks function were generated for the 11 carriers and common filters of 7 images (shown in Figure 6) and those generated from the differential evolution were used to recover the surface. In the end, the quadratic error between the recovered surface and the simulated surface with the peaks function was calculated. Three comparison experiments were carried out. Experiment 1, without adding noise to the images. Experiment 2, adding random noise of 0.01% to component a; and experiment 3, adding random noise of 0.01% to component ω_0 of the images.

In Figure 6 of the comparison filters, ω_0 corresponds to the carrier value for which they were designed, the diagram shows the Fourier α cut-off points of the filter (angles and the direction is clockwise as it is customary to illustrate for this type of filters).

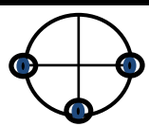
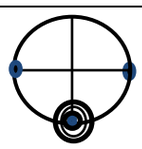
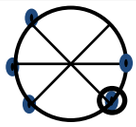
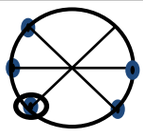
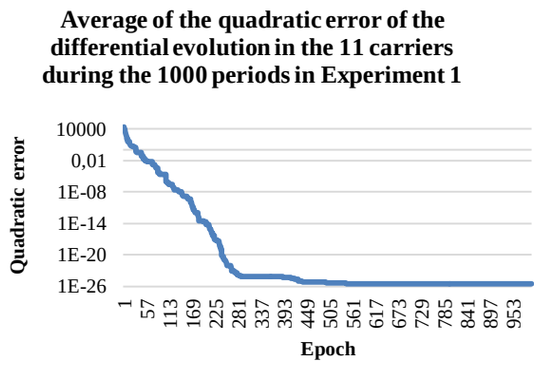
Number of filters	$\omega_0 t$	Diagram	Coefficients
1	$\frac{\pi}{2}$		$\frac{[1,0,-3,0,3,0,-1]}{[0,2,0,-4,0,2,0]}$
2	$\frac{\pi}{2}$		$\frac{[-1,0,7,0,-7,0,1]}{[0,-4,0,8,0,-4,0]}$
3	$\frac{\pi}{3}$		$\frac{\sqrt{3}[-1,3,3,0,-3,-3,1]}{3[0,2,0,-4,0,2,0]}$
4	$\frac{\pi}{3}$		$\frac{\sqrt{3}[0,1,1,0,-1,-1,0]}{[-1,-1,1,2,1,-1,-1]}$

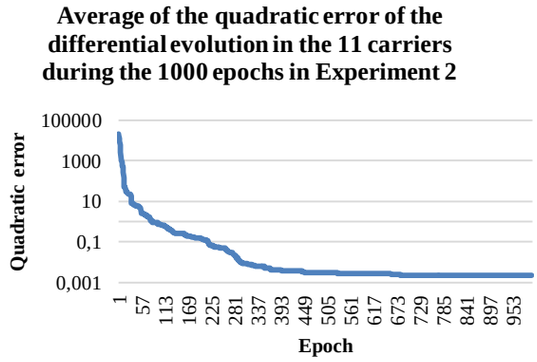
Figure 6 Common filters of 7 images

Results

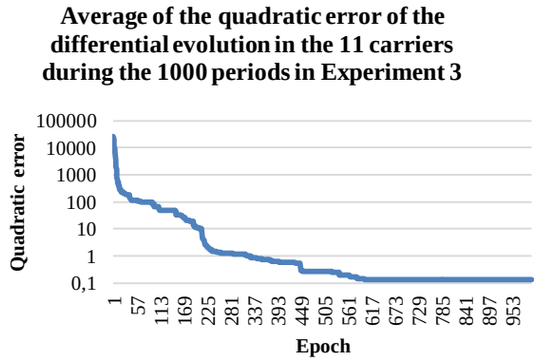
In Graph 1, Graph 2, and Graph 3 the average of the quadratic error of the differential evolution is shown for the 11 carriers evaluated in the 3 experiments (the quadratic error is expressed in base 10 logarithmic scale), a similar behavior is observed, as well as the fact that the error starts too high converging early in the period at a value close to 0.



Graph 1 Average of the quadratic error of the differential evolution in the 11 carriers during the 1000 periods in Experiment 1
Source: Prepared by the authors



Graph 2 Average of the quadratic error of the differential evolution in the 11 carriers during the 1000 periods in Experiment 2
Source: Prepared by the authors



Graph 3 Average of the quadratic error of the differential evolution in the 11 carriers during the 1000 periods in Experiment 3
Source: Prepared by the authors

Table 1 and Table 2 are the results of experiment 1. Table 1 shows the individuals selected by the differential evolution algorithm for each carrier of the experiment. These individuals contain the $\alpha_1, \alpha_2, \alpha_3$, and α_4 that define the filter.

ISSN 2444-5002

$\omega_0 t$	α_1	α_2	α_3	α_4
$\frac{\pi}{10}$	3.39341494	0.31415784	0.31416257	0.31416178
$\frac{\pi}{10}+\frac{2\pi}{25}$	0.56584501	0.56548641	0.56548579	3.68985352
$\frac{\pi}{10}+\frac{4\pi}{25}$	0.81684604	0.81681588	3.75344502	0.81680483
$\frac{\pi}{10}+\frac{6\pi}{25}$	3.44769663	1.06814547	1.06814117	1.06810668
$\frac{\pi}{10}+\frac{8\pi}{25}$	1.31944629	1.31950415	1.31891933	1.31880409
$\frac{\pi}{2}$	1.56295155	1.57069825	1.57079604	1.56978077
$\frac{\pi}{10}+\frac{12\pi}{25}$	1.82213	1.82210915	1.82213792	1.5884965
$\frac{\pi}{10}+\frac{14\pi}{25}$	2.073	2.073454	2.07345115	0.01856206
$\frac{\pi}{10}+\frac{16\pi}{25}$	2.32477803	0.00089924	2.32478016	2.32494418
$\frac{\pi}{10}+\frac{18\pi}{25}$	2.57611004	0.00094596	2.57609418	2.57611317
$\frac{9\pi}{10}$	2.82743408	0.20611196	2.82747649	2.82743618

Table 1 Values $\boldsymbol{\alpha_1}$, $\boldsymbol{\alpha_2}$, $\boldsymbol{\alpha_3}$, and $\boldsymbol{\alpha_4}$ in Experiment 1
Source: Prepared by the authors

It is observed how many of the α values obtained by the differential evolution correspond to the carrier value, others to values close to 0, and others to values that correspond to harmonics of the carrier.

$\omega_0 t$	Quadratic error Filter obtained	Quadratic error Filter 1	Quadratic error Filter 2	Quadratic error Filter 3	Quadratic error Filter 4
$\frac{\pi}{10}$	2.0267E-26	1289.17545	709.193493	1011.83758	2608.92567
$\frac{\pi}{10}+\frac{2\pi}{25}$	6.0748E-28	495.490852	62.8168649	91.3724312	483.399975
$\frac{\pi}{10}+\frac{4\pi}{25}$	3.1399E-28	149.023385	4.27249863	2.9615834	55.3055054
$\frac{\pi}{10}+\frac{6\pi}{25}$	1.3602E-28	28.4029947	0.13122533	0.00014448	0.23972935
$\frac{\pi}{10}+\frac{8\pi}{25}$	1.8345E-28	1.73799776	0.00044905	3.25728372	20.5986079
$\frac{\pi}{2}$	1.8241E-28	2.4351E-28	**1.7947E-28**	38.0972638	33.5218508
$\frac{\pi}{10}+\frac{12\pi}{25}$	2.023E-28	1.73799776	0.00044905	167.890785	20.5986079
$\frac{\pi}{10}+\frac{14\pi}{25}$	3.9314E-28	28.4029947	0.13122533	509.098423	0.23972935
$\frac{\pi}{10}+\frac{16\pi}{25}$	1.1446E-27	149.023385	4.27249863	1303.08776	55.3055054
$\frac{\pi}{10}+\frac{18\pi}{25}$	8.6198E-27	495.490852	62.8168649	3139.45571	483.399975
$\frac{9\pi}{10}$	2.9105E-25	1289.17545	709.193493	7858.36742	2608.92567

Table 2 Quadratic error between the original surface and that obtained by the filters in Experiment 1
Source: Prepared by the authors

Table 2 contains the quadratic error between the simulated original surface of the peaks function and the surfaces recovered from experiment 1. It is observed that the filter obtained for each carrier generally obtained better results than the other filters compared here, the filter error value that was lower for each carrier is shown in bold (the same is shown for Table 4 and Table 6).

Tables 3 and 4 show the results of experiment 2. Adding a noise level of 0.01% to component a of the images. Table 3 shows the values $\alpha_1, \alpha_2, \alpha_3$ and α_4 obtained in the differential evolution. It can be noted how the values differ more from the carrier than in experiment 1.

$\omega_0 t$	α_1	α_2	α_3	α_4
$\frac{\pi}{10}$	0.31415927	3.84595269	4.3097081	3.21293577
$\frac{\pi}{10}+\frac{2\pi}{25}$	0.56408641	3.47891916	3.47891919	3.47891914
$\frac{\pi}{10}+\frac{4\pi}{25}$	0.81681818	0.68943062	0.81683439	4.6475956
$\frac{\pi}{10}+\frac{6\pi}{25}$	0.42835963	0.42835958	0.42835953	0.97643749
$\frac{\pi}{10}+\frac{8\pi}{25}$	4.34392235	1.31946884	0.24623126	1.31946627
$\frac{\pi}{2}$	2.88707736	2.88707727	1.58142495	2.88707767
$\frac{\pi}{10}+\frac{12\pi}{25}$	1.82214712	0.08621156	4.86019418	1.8220794
$\frac{\pi}{10}+\frac{14\pi}{25}$	4.66441939	2.07345217	3.32705553	2.07345113
$\frac{\pi}{10}+\frac{16\pi}{25}$	2.32477856	0.03785148	2.61020667	4.95782448
$\frac{\pi}{10}+\frac{18\pi}{25}$	2.37501618	2.37501623	5.39835857	2.37501571
$\frac{9\pi}{10}$	1.02587409	0.76972043	2.82743339	4.36500391

Table 3 Values $\boldsymbol{\alpha_1}$, $\boldsymbol{\alpha_2}$, $\boldsymbol{\alpha_3}$, and $\boldsymbol{\alpha_4}$ in Experiment 2
Source: Prepared by the authors

And Table 4 shows the quadratic error between the original surface peaks and the surface obtained by the filters generated in experiment 2. It is observed that it obtains better results in general for 7 values of the carrier, so this procedure reflects good results against 0.01% noise in component a of the images.

$\omega_0 t$	Quadratic error Filter obtained	Quadratic error Filter 1	Quadratic error Filter 2	Quadratic error Filter 3	Quadratic error Filter 4
$\frac{\pi}{10}$	**11.3156004**	1332.04539	829.635101	1052.93484	2682.98442
$\frac{\pi}{10}+\frac{2\pi}{25}$	**1.85225274**	536.444531	44.4973128	73.8519551	438.797349
$\frac{\pi}{10}+\frac{4\pi}{25}$	**2.06810974**	145.131457	4.93457309	3.27576934	56.2346913
$\frac{\pi}{10}+\frac{6\pi}{25}$	27.3841552	31.1390406	2.3100546	**0.0915791**	0.50409375
$\frac{\pi}{10}+\frac{8\pi}{25}$	**0.12627019**	1.65109314	0.37787341	4.75278043	18.1970708
$\frac{\pi}{2}$	0.48040327	**0.00288633**	0.01147011	37.0954392	35.5608772
$\frac{\pi}{10}+\frac{12\pi}{25}$	0.48816892	1.56377579	**0.0331104**	163.054158	23.6688965
$\frac{\pi}{10}+\frac{14\pi}{25}$	70.2612303	27.8119454	2.82438951	5753.21601	5556.94097
$\frac{\pi}{10}+\frac{16\pi}{25}$	**0.59793114**	154.792944	4.11593077	1228.03868	49.9164952
$\frac{\pi}{10}+\frac{18\pi}{25}$	**5.3790476**	475.695833	73.4079079	2989.77769	449.976993
$\frac{9\pi}{10}$	**1.41064121**	1262.87099	747.974768	8123.44365	2767.71522

Table 4 Quadratic error between the original surface and that obtained by the filters in Experiment 2
Source: Prepared by the authors

ISSN 2444-5002

Tables 5 and 6 show the results of experiment 3, where noise is added to the images in component ω_0. Table 5 contains the corresponding values $\alpha_1, \alpha_2, \alpha_3$ and α_4 obtained. Table 6shows the quadratic errors for experiment 3. Table 5 shows how the α values obtained are varied, trying to compensate for the error in ω_0. Table 6 shows how the filter obtained generates bad approximations of the original surface with the filters created. Although the other filters also do not have considerably good results in this type of noise.

$\omega_0 t$	α_1	α_2	α_3	α_4
$\frac{\pi}{10}$	3.32920506	1.30797008	0.18626818	4.53170912
$\frac{\pi}{10}+\frac{2\pi}{25}$	1.04400711	0.41752438	1.48632945	3.59359454
$\frac{\pi}{10}+\frac{4\pi}{25}$	2.41891002	4.67177629	0.84677996	4.77445194
$\frac{\pi}{10}+\frac{6\pi}{25}$	3.78062461	1.34567746	1.7207E-15	1.34567746
$\frac{\pi}{10}+\frac{8\pi}{25}$	6.09515811	1.30248931	4.52554356	2.21220134
$\frac{\pi}{2}$	1.13518769	5.09820149	1.13518767	1.13518751
$\frac{\pi}{10}+\frac{12\pi}{25}$	1.79390759	1.03679378	3.38555486	4.90012928
$\frac{\pi}{10}+\frac{14\pi}{25}$	1.87011461	3.53110598	1.83660486	5.26091338
$\frac{\pi}{10}+\frac{16\pi}{25}$	0.08522184	2.29078968	0.08522179	6.28318531
$\frac{\pi}{10}+\frac{18\pi}{25}$	1.31637597	1.31637598	2.51937157	1.31637604
$\frac{9\pi}{10}$	0.32691101	0.21442545	2.82578976	5.31039657

Table 5 Values α_1, α_2, α_3, and α_4 in Experiment 3
Source: Prepared by the authors

ω_0t	Quadratic error Filter obtained	Quadratic error Filter 1	Quadratic error Filter 2	Quadratic error Filter 3	Quadratic error Filter 4
$\frac{\pi}{10}$	**372.645432**	2615.21151	1367.18855	1313.2444	2966.18916
$\frac{\pi}{10}+\frac{2\pi}{25}$	**31.7460575**	726.451404	42.8653768	59.8989512	409.622363
$\frac{\pi}{10}+\frac{4\pi}{25}$	82.319874	224.439255	22.1017502	**13.3538321**	36.6120774
$\frac{\pi}{10}+\frac{6\pi}{25}$	25.9686161	36.6630661	**25.8437282**	36.6350729	37.7581843
$\frac{\pi}{10}+\frac{8\pi}{25}$	35.3537115	**2.3875674**	4.09725675	21.6537373	56.5979154
$\frac{\pi}{2}$	22.3487838	5.15276423	**1.42260927**	109.567887	5.87751501
$\frac{\pi}{10}+\frac{12\pi}{25}$	101.036962	7.01629465	**0.58385936**	287.474212	35.3712173
$\frac{\pi}{10}+\frac{14\pi}{25}$	28.6485437	26.1559882	**10.0363416**	42613.6722	30350.6701
$\frac{\pi}{10}+\frac{16\pi}{25}$	29.7515912	109.889506	**9.05457366**	1824.03349	693.502067
$\frac{\pi}{10}+\frac{18\pi}{25}$	**7.123764**	607.816808	44.8672903	6808.00038	2894.14073
$\frac{9\pi}{10}$	**41.6824643**	2035.08558	459.75393	10437.1562	4114.31313

Table 6 Quadratic error between the original surface and that obtained by the filters in Experiment 3
Source: Prepared by the authors

Conclusions

In this paper a different way to the state of the art was tested in the use of soft computing for the technique of phase shifting interferometry. We looked for filters that are a treatable solution at different noise levels in obtaining image captures. Based on the results section, the differential evolution seems to obtain satisfactory filters for noise of 0.01% in the a component of the model of the images, while it does not obtain good results against noise of 0.01% in the component ω_0 of the captured images. It was also observed that the periods in which the differential evolution reaches values close to 0 are early, so it is inferred that it has been a problem not so complicated to solve for the technique or that it has stagnated in local optimums. This method could be improved and analyzed by evaluating other noise scenarios or using a larger filter search space, that is, with more images.

References

Brownlee, J. (2011). Clever Algorithms Nature-Inspired Programming Recipes.

Geng, J. (2011). Structured-light 3D surface imaging: a tutorial. (O. S. America, Ed.) Advances in Optics and Photonics, 128-160.

Gutiérrez García, J. C., Gutiérrez García, T. A., Mosiño, J. F., Vázquez Domínguez, E., Martínez, A., & Arroyo, C. J. (2015). A novel application of the white light/fringe projection duo: recovering high precision 3-D images from fossils for the digital preservation of morphology. Paleontología Electronica , 1-13.

Gutiérrez García, J. C., Mosiño, J. F., Martínez, A., Gutiérrez García, T. A., Vázquez Dominguez, E., & Arroyo Cabrales, J. (2013). Practical eight-frame algorithms for fringe projection profilometry. Optics Express, 21(1), 903-917.

Mosiño, J. F., Gutiérrez-García, J. C., Gutiérrez-García, T. A., Castillo, F., García-González, M. A., & Gutiérrez-García, V. A. (2011). Algorithm for phase extraction from a set of interferograms with arbitrary phase shifts. Optics Express, 19(18), 4908-4923.

ISSN 2444-5002

Mosiño, J. F., Malacara Doblado, D., & Malacara Hernández, D. (2009). Calculus of exact detuning phase shift error in temporal phase shifting algorithms. Optics Express, 17(18), 15766-15771.

Ordoñes Nogales, S. (2016). Algoritmo Iterativo Eficiente para el Análisis de Interferogramas con Corrimiento de Fase Aleatorio. Universidad de Guadalajara, Centro Universitario de Ciencias Exactas e Ingenierías, Guadalajara.

Simon, D. (2013). Evolutionary Optimization Algorithms. New Jersey: John Wiley & Sons, Inc.

Zuo, C., Huang, L., Zhang, M., Chen, Q., & Asundi, A. (6 de Mayo de 2016). Temporal phase unwrapping algorithms for fringe projection profilometry: A comparative review. Optics and Lasers in Engineering, 84-103.

ISSN 2444-5002