Fringe pattern demodulation using one-dimensional continuous Morlet and Paul wavelet transforms

This software uses the continuous one-dimensional Morlet wavelet transform to extract the phase of a fringe pattern. This algorithm processes the fringe pattern image on a row-by-row basis and is explained in detail in the following papers [1, 2].

Software Parameters

The parameters for this software are as follows:

Fringes:
This is a one-dimensional array that contains the fringe pattern image stored as BYTE data type format. The image is scanned on a row-by-row basis and is then stored in the fringes array. The direction of scanning should be perpendicular to the direction of fringes.

Use_FFT
This variable should be set to YES if you would like the software to calculate the wavelet transform using the frequency domain. To calculate the wavelet transform using convolution in the time domain, set **use_FFT** to NO.

Image width and image height
These are the dimensions of the fringe pattern image in pixels. The WFA function uses the Fourier transform to compute the wavelet transform. When the width of the image is 2^n where n is a positive integer, the WFA uses the fast Fourier transform; otherwise it uses the discrete Fourier transform. This means that the computation for the WFA function is performed much more quickly when the width of the image is set to be a power of 2.

to_extend_image
Set this variable to YES if you would like the software to extend the fringe pattern borders. The left and right borders of the image are extended using the linear predictive algorithm [3]. If you decided to calculate the wavelet transform in the time domain, you then determine the number of pixels to extend the image using the variable `extend_by_pixels`.

If you chose to calculate the wavelet transform in the frequency domain, then the `extend_by_pixels` variable will be calculated automatically by the software to set the width of the image to 2^n.

starting_scale, scale_step, ending_scale
The starting scale, scale step and ending scale parameters could be set to the default values 1, 1, 300 respectively. This determines the number of daughter Morlet wavelets that will used to extract the phase of the fringe pattern, which is given by

$$number\ of\ daughter\ wavelets\ = \frac{ending\ scale - starting\ scale}{scale\ step}$$

The required computation time increases linearly as the number of daughter wavelets used increases. To improve the computation speed and the performance of the algorithm, the following guidelines could be followed:

The WFA function scans a fringe pattern on a row by row basis in order to extract its phase components. Since the scanning direction is perpendicular to the direction of the fringes, a row crosses a certain number of fringes. A period of a fringe is defined as the number of pixels in a row that crosses this fringe. The image could contain a number of fringes with different periods.

starting scale: This parameter must be set to a value larger than 2. The approximate optimal value for this parameter is the minimum period of a fringe in the image minus 5.

ending scale: This parameter must be set to a value larger than 2. The approximate optimal value for this parameter is the maximum period of a fringe in the image plus 5.

scale step: This parameter could be set according to the following equation

Scale step ≅ (ending scale – starting scale)/50

This allows the wavelet transform to use 50 daughter wavelets to analyze an image. As a rule of thumb, the scale step should be set to a small value (i.e., > 0.25) when the phase variations in the image are large and when the period of the fringes is small (period < 10 pixels).

If you would like the algorithm to work automatically, set the starting_scale to 5, the scale_step to 3 and the ending_scale to 200.

Ridge algorithm: The WFA function uses either the maximum algorithm or the cost function algorithm to extract the ridge. The value of this parameter must be set to MAX or COST. When this parameter is set to MAX, the WFA function uses the maximum ridge extraction method; when is set to COST, it uses the cost function ridge extraction method.

The maximum ridge extraction technique should be used in most cases and it requires less computation time than the cost function method, which should be used in special cases only. For example, when analyzing noisy images, when the phase in a fringe pattern varies quickly or when the maximum ridge extraction method does not give satisfactory results. The default value for this parameter should therefore be set to 1.

Sigma: This parameter controls the width of the envelope of Morlet wavelet and it should be set to 1. In some cases where the fringe pattern is noisy and the phase variations in the image are slow, this parameter can be set to a value between 0.5 and 1. In the cases where the phase in the image varies quickly, the value of the sigma parameter could be to a value between 1 and 2. Also this parameter could be set between 1 and 2 in cases where there are less than seven fringes in the image.

Mother wavelet: Paul and Morlet
This software supports two mother wavelets that can be used to extract the phase of fringe patterns: the complex Paul wavelet and complex Morlet wavelet. The Paul wavelet should be used if the signal to noise ratio of a fringe pattern is high; whereas the Morlet wavelet should be used when the signal to noise ratio is low.

The user can integrate other mother wavelets into this software by adding them from the **file list_of_wavelets.c** distributed with this software.

WFA versus WTA

This software provides the functions WFA and WTA which are capable of extracting the phase of tilted fringe patterns (i.e., fringe patterns that contain a spatial carrier and do not contain closed fringes). Both functions are mathematically completely equivalent, but the WFA function uses the frequency domain to implement the wavelet transform; whereas the WTA utilizes the time domain to perform the wavelet computation.

The WFA function is faster in terms of execution time. For example: processing a fringe pattern with the dimensions of 512 × 512 pixels requires approximately one second. Also it has the ability to remove the background illumination components from a fringe pattern automatically.

The execution times stated here were measured using a Pentium 4 computer, with 3GHz clock speed, 4G Byte RAM and using windows XP operating system platform.

The WTA function is slower to execute than the WFA function. For example: processing a fringe pattern with dimensions of 512 × 512 pixels requires approximately one minute. The WTA function is not capable of automatically removing the background illumination of a fringe pattern. A simple method to remove the background illumination in this case is to subtract the mean of each row in a fringe pattern from that row. This method is implemented in the software prior processing the fringe patterns using the WTA function.

The WTA function has better ability in processing the borders of a fringe pattern. So if the borders of a fringe pattern contain important information then it is advisable to use the WTA function in this case.

References

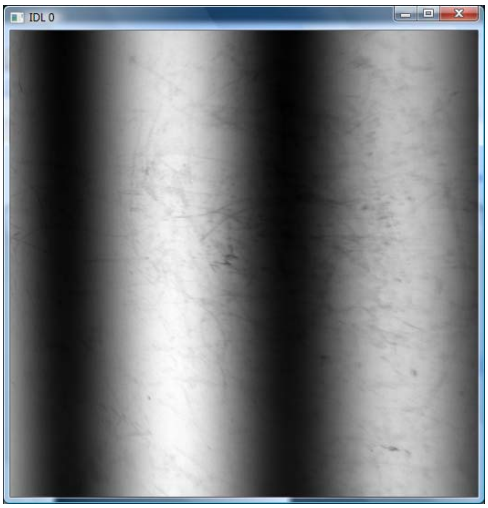
[1] Abdulbasit Z. Abid, Munther A. Gdeisat, David R. Burton, Michael J. Lalor, Hussein S. Abdul-Rahman, and Francis Lilley, “Fringe pattern analysis using a one-dimensional modified Morlet continuous wavelet transform,” SPIE Vol. 7000, April 2008.

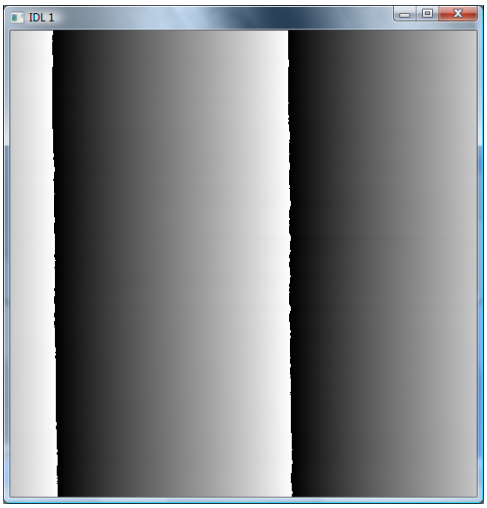
[2] Munther A. Gdeisat, David R. Burton and Michael J. Lalor, ""Spatial Carrier Fringe Pattern Demodulation Using a Two-Dimensional Continuous Wavelet Transform,” Applied Optics, Vol. 45, No. 34, pp. 730-743, 2006.

[3] Chapter 13.6, Numerical recipes in C, second edition (1992), Cambridge Press.

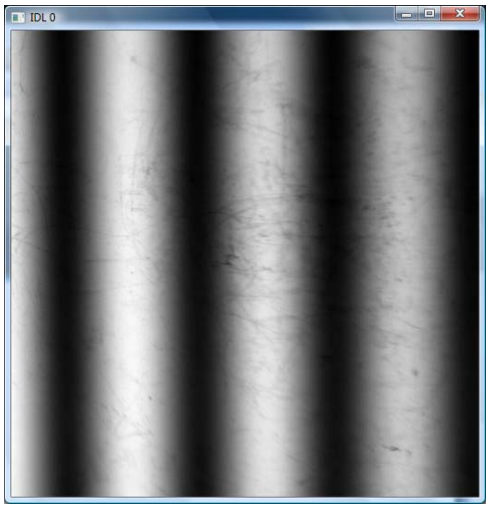
Fringe patterns processed using the wavelet transform

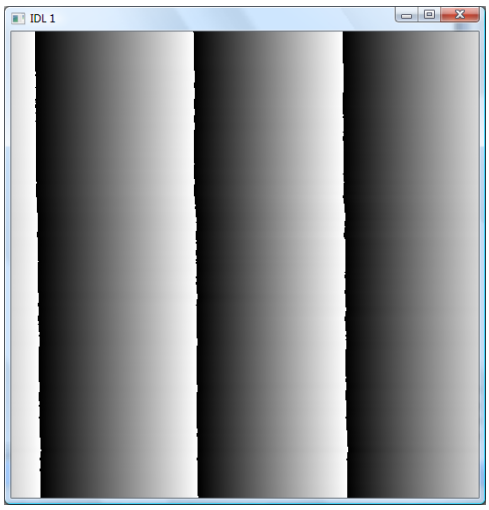
Starting scale = 200
Ending scale = 300
Scale step = 5
Sigma = 1.5
Ridge algorithm = MAX
Image size = 512 *X* 512



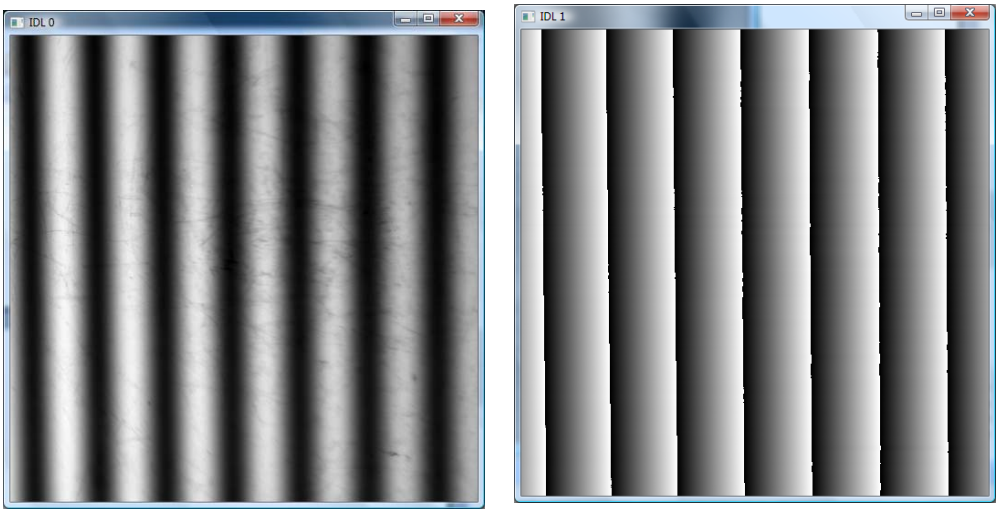


Starting scale = 80
Ending scale = 150
Scale step = 3
Sigma = 1.5
Ridge algorithm = MAX
Image size = 512 *X* 512

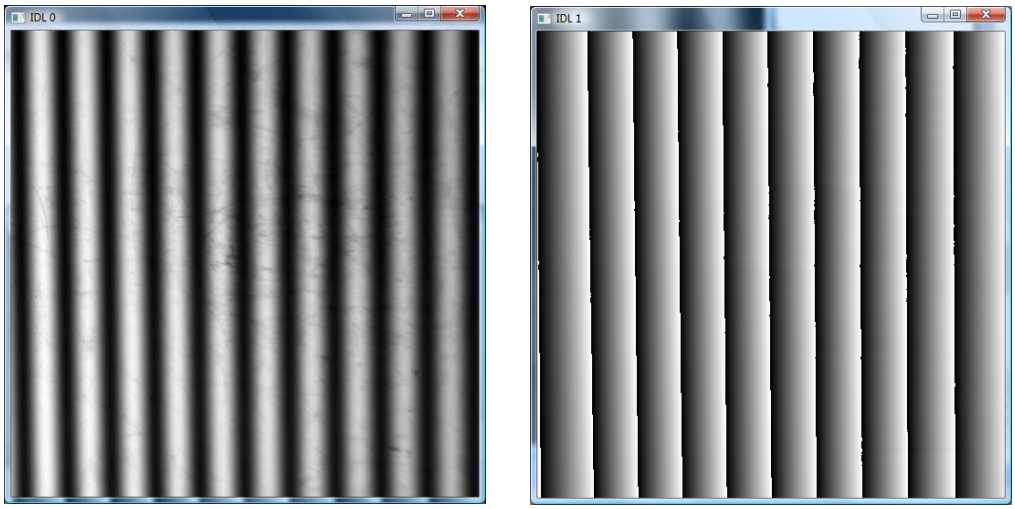




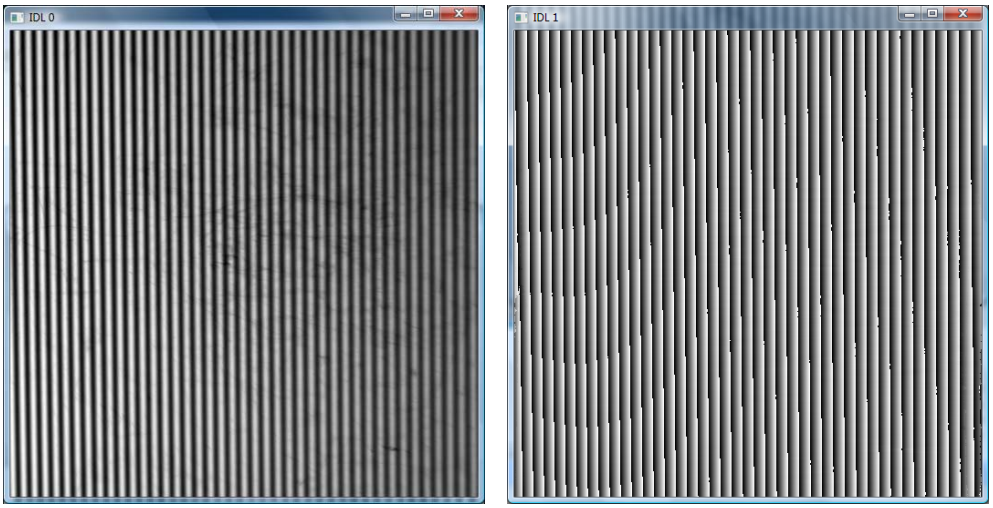
Starting scale = 60
Ending scale = 80
Scale step = 1
Sigma = 1
Ridge algorithm = MAX
Image size = 512 *X* 512



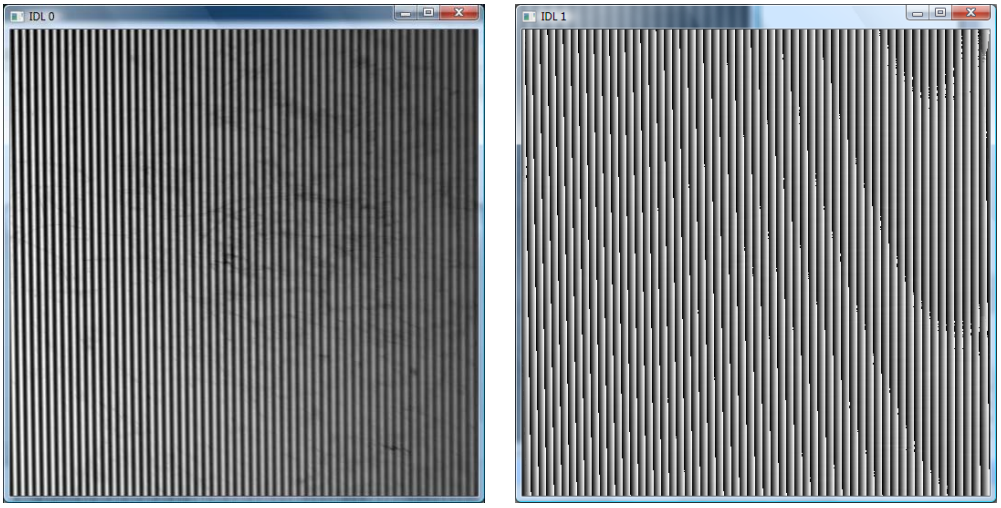
Starting scale = 40
Ending scale = 60
Scale step = 1
Sigma = 1
Ridge algorithm = MAX
Image size = 512 *X* 512



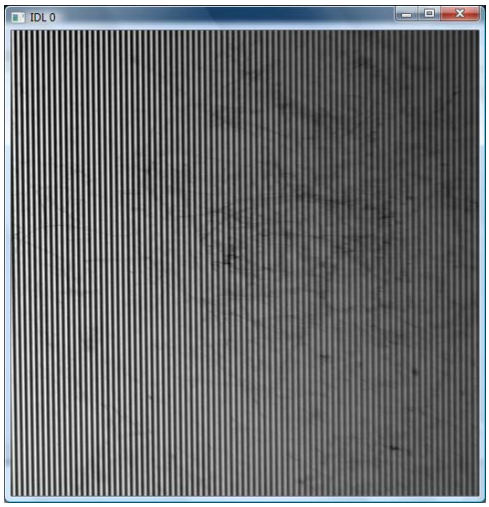
Starting scale = 5
Ending scale = 20
Scale step = 0.5
Sigma = 1
Ridge algorithm = MAX
Image size = 512 *X* 512

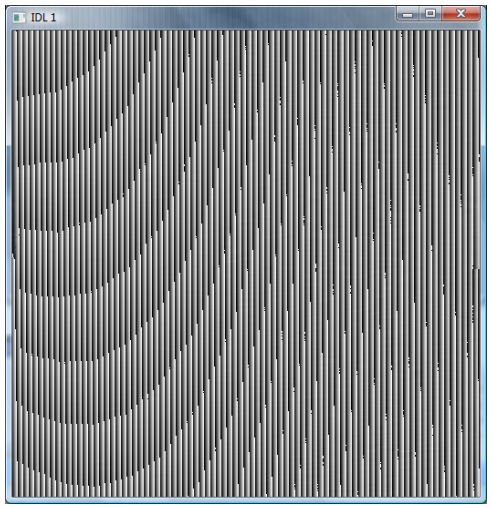


Starting scale = 5
Ending scale = 20
Scale step = 0.5
Sigma = 1
Ridge algorithm = MAX
Image size = 512 *X* 512

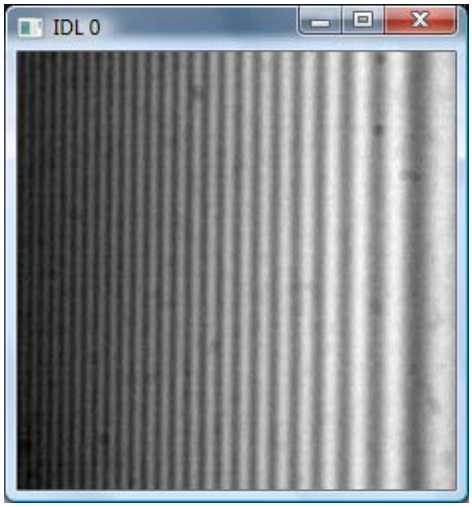


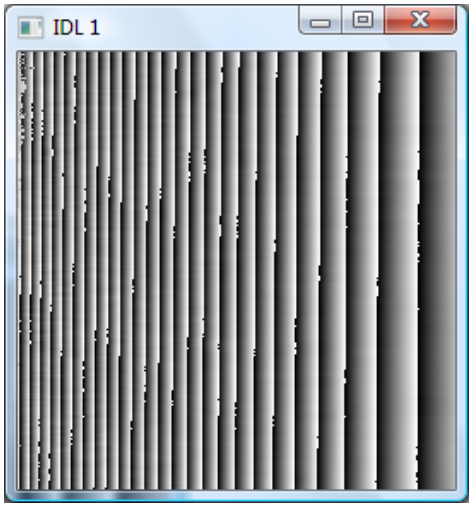
Starting scale = 3
Ending scale = 10
Scale step = 0.25
Sigma = 1
Ridge algorithm = MAX
Image size = 512 *X* 512



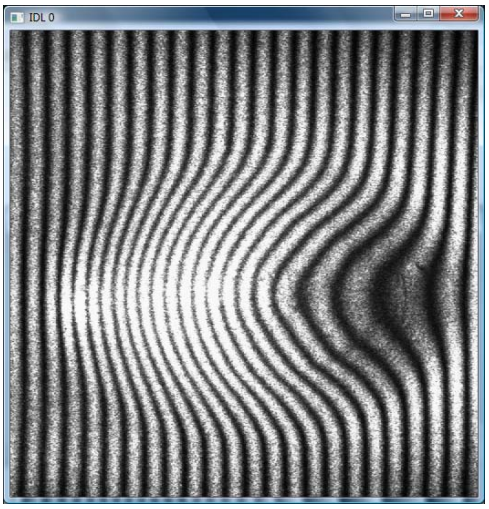


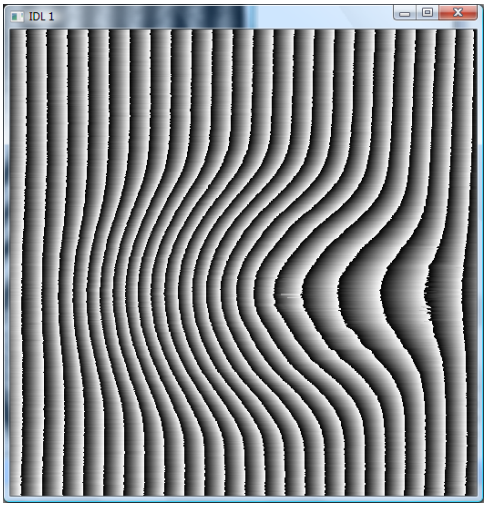
Starting scale = 2
Ending scale = 25
Scale step = 0.1
Sigma = 1.5
Ridge algorithm = COST
Image size = 256 *X* 256





Starting scale = 8
Ending scale = 60
Scale step = 0.5
Sigma = 1.5
Ridge algorithm = COST
Image size = 512 *X* 512





Starting scale = 5
Ending scale = 50
Scale step = 1
Sigma = 1.5
Ridge algorithm = COST
Image size = 512 *X* 512