The GridMethod toolbox - v1.0

May 30, 2018

Abstract

This document gives guidelines for installing and using the GridMethod toolbox, a collection of MATLAB codes for measuring displacement and strain fields on the surface of a deformed specimen on which a regular grid has been deposited.

The current version (v1.0) of the toolbox has been tested under the following systems:

- MATLAB *R2015b with* GCC *5.2.1 on Linux Ubuntu 15.10 64 bits;*
- MATLAB *R2015b with* XCODE CLANG++ *on Mac OS 10.11.3. - See patch in Section 2.2.*

Important note: The GridMethod toolbox can be used for non-profit academic research only. The codes `script_gridmethod.m`, `script_generate_synthetic_grids.m`, `build_window.m`, `LSA.m`, `register_gridimages.m`, and `calculate_U_EPS.m` are distributed under the terms of the GNU GPL.

Authors: The GridMethod toolbox has been developed by Frédéric SUR, Michel GRÉDIAC and Benoît BLAYSAT.

1 How to cite this work?

Anyone finding the GridMethod toolbox useful is kindly asked to cite the review paper [3] and reference [1], in which compensating the movement of the grid between current and reference grid images is proposed for the first time. For specific aspects of the grid method, please cite the appropriate corresponding papers (see [3] for a comprehensive review).

2 Installation

2.1 Unzipping the archive

Unzip the archive `gridmethod.tgz`. The files are in three directories: the scripts are in `scripts` (Section 3), the functions are in `functions` (Section 4), and the example grid images are in `data` (Section 5). The archive also contains the present documentation, the GNU General Public Licence, and the terms and conditions for using 2D phase unwrapping from Liverpool John Moores University (see Section 2.2).

2.2 Setting up 2D phase unwrapping

The directory `functions` contains the (unchanged) source code of 2D-SRNCP to be compiled with `mex`. It is available at the following URL:
`https://www.ljmu.ac.uk/research/centres-and-institutes/faculty-of-engineering-and-technology-research-institute/geri/phase-unwrapping`
This code implements 2D phase unwrapping as described in [5].

If `mex` is correctly configured together with a C++ compiler in the user's MATLAB installation, he/she should be able to compile this `mex` file by typing in MATLAB's command window, in the `functions` directory:

```
>> mex unwrap2D.cpp
```

Note that the `unwrap2D` function can be replaced with any 2D phase unwrapping function, without any modification in the remainder of the codes as long as the new function satisfies the syntax described in Section 4.3.

For Mac OS users: the `cpp` file has to be patched by:

- Including `malloc.h` by `#include <malloc/malloc.h>`
- Adding `return no;` at the end of function `find_pivot`.

The patched file is `unwrap2D_mac.cpp`. Mac OS users should rename this file as `unwrap2D.cpp` (save the original file elsewhere) and compile with `mex` as described above.

2.3 Setting MATLAB'S path

To use the files within MATLAB, simply add the three directories `data`, `scripts`, and `functions` to MATLAB's path:

- either with a right-click on the folder name in *Current Folder* followed by *Add to Path / Selected Folder*;
- or by typing the following line in the MATLAB's command window in the root directory of the toolbox:

```
>> addpath data functions scripts
```

3 Scripts

Scripts illustrate the use of the GridMethod toolbox. The user is asked to run the scripts step-by-step and read the comments in the corresponding MATLAB files.

3.1 `script_grid_method`

This script loads two grid images (in `data`), builds the analysis window, estimates and unwraps the phases, registers the grid images (to solve the 2π indetermination in phases), and eventually calculates displacement, strain, and rotation maps. Intermediate displays are available.

Five experiments are available, the user just needs to change variable `experiment` before running the script. Variable `method_register` allows the user to choose between the two methods available in `register_gridimages`. Variable `procedure` allows to choose between the two procedures for calculating displacement and strain maps.

3.2 `script_generate_synthetic_grids`

This script generate synthetic grid images. Synthetic phase maps modeling a grid deformation are defined (and saved in the `data` directory), and two synthetic grid images are generated: one for the reference state, the other for the deformed state. The output grid images (saved in `data`) are quantized over 12 bits, the final output being multiplied by 2^4 to mimic the output of a Sensicam QE camera.

The grid model is as follows:

$$s(x,y) = \frac{A}{2}\Big(2 + \gamma \cdot \ell(2\pi f x + \phi_1(x,y)) + \gamma \cdot \ell(2\pi f y + \phi_2(x,y))\Big)$$

where:

- $A > 0$ is the field illumination;
- $\gamma \in [0,1]$ is the contrast of the oscillatory pattern;
- the line profile ℓ is a 2π-periodic real function with a peak amplitude equal to 1 and average value 0;
- f is the nominal frequency of the carrier, the grid pitch being $p = 1/f$;
- $\phi_1(x,y)$ and $\phi_2(x,y)$ are the carrier phase modulations due to specimen surface displacements along the x and $y-$axes respectively.

In the script, $\ell(x) = \sin^3(x)$ to mimic sharp lines and deliberately induce harmonics. The values of A, γ and p as well as the phase maps can be edited.

Sensor noise can be added to the ideal grids, either as a Gaussian white noise, or as a realistic signal-dependent noise (whose parameters are estimated in [6]). The user should change the value of the variable `type_noise`.

4 Functions

4.1 `build_window`

```
function g = build_window( profile, sigma )
% build window analysis
% 1) input
% profile : 0=Gaussian, 1=bi-triangular, 2=triangular-rectangular, 3=bi-rectangular
% sigma: 'half-width' of the analysis window
% 2) output
% g: analysis window
```

This function builds the analysis window. For the Gaussian window, use σ (standard deviation) larger than or equal to the grid pattern pitch [7]. For the bi-triangular window, use σ equal to an integer multiple of the grid pattern pitch and for the triangular-rectangular and birectangular window, use σ equal to an integer multiple of half of the grid pattern pitch [9]. The influence of the analysis windows on the metrological performance is discussed in [9]. The conclusion is that Gaussian windows should be used except in specific situations when very localized phenomena are to be characterized and the noise level is low. The bi-triangular window should be used in such situations.

4.2 `LSA`

```
function [ PHI_X, PHI_Y, modX, modY ] = LSA( im, g, p, theta)
% Local spectrum analysis (LSA)
% usage:
% 1) input:
% im: grid image (double)
% g: analysis window (cf build_window.m)
% p: grid pattern pitch (in pixels)
% theta : grid angle
% 2) output
% PHI_X: phase along x-direction
% PHI_Y: phase along y-direction
% modX: modulus of the LSA along x-direction
% modY: modulus of the LSA along y-direction
```

This function implements *localized spectrum analysis* (LSA), proposed in [1, 2] after [10], using the fast Fourier transform (FFT) to speed up the calculation. It gives an estimation of the grid phase modulations. This procedure relies on the windowed Fourier transform employed with one frequency only, namely the nominal frequency of the grid. The grid can be rotated with an angle θ (in radian) with the camera sensor.

4.3 `unwrap2D`

Usage:

```
>> phase_unwrapped = unwrap2D(single(phase));
```

This functions (compiled `mex` file) implements the algorithm of [5], as mentioned in Section 2.2. Note that `unwrap2D` needs a single precision input. This function can be replaced by any 2D phase unwrapping algorithm.

4.4 `register_gridimages`

```
function [ k, l ] = register_gridimages( method, data, p, imgrid1, imgrid2, ...
PHI1_X, PHI1_Y, PHI2_X, PHI2_Y, theta )
% compensate the uniform translation of 2 pi x k from PHI1_Y to PHI2_Y
% and of 2 pi x l between PHI1_X to PHI2_X
% 1) input:
% method: 1 = hand-picked points, 2 =  by cross-correlation on the image grids
% data: area on which the NCC is performed (not containing a hole or a
% crack) in method 1, or coordinates of the two corresponding points in
% theta: grid angle
% method 2
% p: pattern pitch
% imgrid1, imgrid2: grid images
% PHI1_X, PHI1_Y, PHI2_X, PHI2_Y: unwrapped phase maps
% theta: grid angle
% 2) output:
% k,l
```

Phases are estimated modulo 2π. This function aims at compensating the phase translation by registering the grid images through image correlation maximization or hand-picked corresponding points. Grid defects are helpful for registering the grid images with correlation. If the registration process fails, corresponding points have to be selected by hand.

4.5 `calculate_U_EPS`

```
function [ UX, UY, EPSXX, EPSYY, EPSXY, WXY ] = calculate_U_EPS( p, ...
PHI1_X, PHI1_Y, PHI2_X, PHI2_Y, procedure, maxiter)
% calculate displacement and strain components from phase maps
% 1) input:
% p: grid pitch
% PHI1_X, PHI1_Y, PHI2_X, PHI2_Y: phase maps in non-deformed (1)
% and deformed (2) grid images
% procedure: 1 = approximate, 2=PHI2 backdeformed in PHI1 axis,
% maxiter: for procedure 2, maximum number of iterations until relative
% improvement below "stopcrit" value (optional parameter, default: maxiter=50)
% 2) output:
% UX, UY: displacement maps
% EPSXX, EPSYY, EPXY: strain maps
% WXY: rotation map
```

This function gives the estimation of displacement (`UX`, `UY`), strain (`EPSS, EPSYY, EPSXY`), and rotation (`WXY`) fields. Procedures 1 and 2 are described in Sections 4.2.2 and 4.2.4 of [3]. By default (if input argument `maxiter` is not passed to the function) Procedure 2 is run with one iteration, which is often sufficient as discussed in [3]. The maximum number of iterations until a stopping criterion is reached can be set with `maxiter` (a typical value for `maxiter` is 50).

5 Data

We propose several grid images, taken before and after deformation, in the `data` directory. These examples concern cases:

- Experiments #1 and 2: tensile test performed on an open-hole specimen cut in a rolled aluminium alloy sheet. The force applied in the first case is greater than in the second case, and so is the strain level. The grainy texture of the strain maps obtained with Procedures 1 or 2 in Experiment #1 is due to the camera sensor noise which propagates to the strain maps. The amplitude of the noise in these maps becomes negligible compared to the actual strain value reached in Experiment #2. This is the reason why the corresponding strain maps seem smoother in the second case than in the first one. The visual aspect of the displacement fields is smooth in both cases, which aptly shows that for a given spatial resolution, retrieving strain fields is much trickier than retrieving displacement fields. More details on this test are available in [8].
- Experiment #3: four-point notched bending test performed on a specimen made in wood. The displacement and strain maps are measured near the right angle of the notch, where a crack appears at a certain load level. Interestingly, heterogeneities are clearly visible in the transverse strain field ϵ_{yy}; they are due to early and late woods, which exhibit different transverse Young's modulus. See [11] for more details.
- Experiment #4: compression test carried out on an asphalt specimen. Heterogeneities in the strain maps are due to the high stiffness ratio between aggregates and binder, see [4] for more details.
- Experiment #5: u_y distribution is such that the strain has an decreasing frequency from the left to the right. Such a distribution has been proposed in [3] for instance to study the loss of amplitude of the signal as a function of its frequency. u_x features an isolated triangular wave along the vertical midline and is null elsewhere. The grid images are produced by `script_generate_synthetic_grids`.
- Experiment #6:

In Experiments 1-4, we can note with Procedure 1 the presence of:

- spurious fringes in ϵ_{xx} and ϵ_{yy} strain maps. They are caused by small fluctuations of the grid pitch due to printing issues of the grid, which is transferred on the specimen prior to testing;
- localized spots caused by a lack of ink in the grid after transfer. These spots are displayed in pairs, the vertical distance between the spots of each pair being constant. The vertical distance is the real displacement between reference and current images. This displacement is compensated with Procedure 2, which leads the aforementioned parasitic fringes and spots to vanish.

Results obtained with Procedure 2 are similar for any value of the maximum number of iterations. It means that the iterative procedure used in Procedure 2 quickly converges, one iteration being sufficient in practice.

Acknowledgements

Dr. Fournely, Moutou-Pitti and Toussaint are gratefully acknowledged for their help in obtaining some of the grid images leading to the displacement and strain maps discussed here, Dr. Blaysat for his intensive testing of the code, and the authors of [5] for making their code publicly available.

6 Graphical user interface

References

[1] C. Badulescu, M. Grédiac, and J.-D. Mathias. Investigation of the grid method for accurate in-plane strain measurement. *Measurement Science and Technology*, 20(9):095102, 2009.

[2] C. Badulescu, M. Grédiac, J.-D. Mathias, and D. Roux. A procedure for accurate one-dimensional strain measurement using the grid method. *Experimental Mechanics*, 49(6):841–854, 2009.

[3] M. Grédiac, F. Sur, and B. Blaysat. The grid method for in-plane displacement and strain measurement: a review and analysis. *Strain*, 52(3):205–243, 2016.

[4] M. Grédiac and E. Toussaint. Studying the mechanical behaviour of asphalt mixtures with the grid method. *Strain*, 49(1):1–15, 2013.

[5] M.A. Herraez, D.R. Burton, M.J. Lalor, and M.A. Gdeisat. Fast two-dimensional phase-unwrapping algorithm based on sorting by reliability following a noncontinuous path. *Applied Optics*, 41(35):7437–7444, 2002.

[6] F. Sur and M. Grédiac. Sensor noise modeling by stacking pseudo-periodic grid images affected by vibrations. *IEEE Signal Processing Letters*, 21(4):432–436, 2014.

[7] F. Sur and M. Grédiac. Towards deconvolution to enhance the grid method for in-plane strain measurement. *AIMS Inverse Problems and Imaging*, 8(1):259–291, 2014.

[8] F. Sur and M. Grédiac. On noise reduction in strain maps obtained with the grid method by averaging images affected by vibrations. *Optics and Lasers in Engineering*, 66:210–222, 2015.

[9] F. Sur and M. Grédiac. Influence of the analysis window on the metrological performance of the grid method. *Journal of Mathematical Imaging and Vision*, 2016. To be published.

[10] Y. Surrel. *Photomechanics*, volume 77 of *Topics in Applied Physics*, chapter Fringe analysis, pages 55–102. Springer, 2000.

[11] E. Toussaint, E. Fournely, R. Moutou Pitti, and M. Grédiac. Studying the mechanical behavior of notched beams with full-field measurements. *Engineering Structures*, 113(5):277–286, 2016.