OpenLSA: An open-source toolbox for computing full-field displacements from images of periodic patterns

Benoît Blaysat \textsuperscript{a,b,*,} Frédéric Sur \textsuperscript{c}, Thomas Jailin \textsuperscript{a}, Adrien Vinel \textsuperscript{a}, Michel Grédiac \textsuperscript{a}

\textsuperscript{a} Université Clermont Auvergne, CNRS, Clermont Auvergne INP, Institut Pascal, F-63000, Clermont-Ferrand, France
\textsuperscript{b} Institut Universitaire de France, Paris, France
\textsuperscript{c} LORIA, Université de Lorraine, CNRS, INRIA, Nancy, BP 239, F-54506, France

\textbf{A B S T R A C T}

In the experimental mechanics community, full-field measurement techniques have gained popularity over the last few decades, revolutionizing traditional testing procedures for materials and structures. While Digital Image Correlation (DIC) remains the most widely used method, its reliance on randomly patterned surfaces limits its metrological performance, and the iterative calculations required for retrieving displacement and strain fields can be computationally expensive. In recent years, there has been a proposal to use optimal checkerboard patterns instead. Images of such periodic patterns can be processed using a method called Localized Spectrum Analysis (LSA). LSA proposes processing these images in the frequency domain using spectral techniques, which reduces computational costs. This paper presents an open-source LSA software written in Python and illustrates two application cases in experimental mechanics.

\section{1. Motivation and significance}

In experimental mechanics, devices providing full-field kinematic data are nowadays routinely used. They mostly rely on cameras that take images of the surface texture of a specimen, the latter being subjected to thermal and/or mechanical loads. Image processing is then performed to retrieve the displacements at every pixel from the stack of images. Considering planar specimens, such displacement maps make it possible to observe the different mechanisms that operate in the specimen properly. Modeling these mechanisms and identifying the material constitutive behavior become then easier. It is worth noting that the elastic domain is often investigated. This usually leads to strain magnitude below $10^{-3}$. Digital Image Correlation (DIC) is the most common image processing technique used to determine the displacement field from a pair of images [1]. It identifies that displacement field by iteratively minimizing the optical residual. In its simplest form, this quantity represents the squared difference of the gray level distributions estimated over small subsets selected in the reference and current images, the latter image being mapped onto the former through the sought displacement. Numerous software are available, including both open-source versions [2–5] or commercial versions [6–8]. These

\* Corresponding author at: Université Clermont Auvergne, CNRS, Clermont Auvergne INP, Institut Pascal, F-63000, Clermont-Ferrand, France.

\textit{E-mail address:} benoit.blaysat@uca.fr (Benoît Blaysat).

https://doi.org/10.1016/j.softx.2024.101826

Received 26 March 2024; Received in revised form 9 July 2024; Accepted 10 July 2024
Available online 23 July 2024

2352-7110/© 2024 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).
2.2. Brief reminder of LSA basics

2.2.1. Displacement measurement with periodic patterns

The present paragraph presents the modeling of images of checkerboard patterns, possibly deformed. The undeformed checkerboard is assumed to be made of square patterns. Let \( P \) (resp. \( F \)) be the gray level intensities of the reference image (resp. current image). It satisfies at any pixel location \( x \), \( \forall \in [0,1] \),

\[
\Gamma(x) = \text{frng} \left( 2\pi k \cdot x + \phi^a(x) + \text{frng} \left( 2\pi k \cdot x + \phi^b(x) \right) \right),
\]

where “frng” is a 2D periodic function, \( k \) (resp. \( k_0 \)) is the wave vector of the pattern periodicity and \( e^a \) (resp. \( e^b \)) is the unit vector such that \( e^a = k_0 / \|k_0\|, \forall \in [\alpha, \beta] \). With checkerboard patterns, the directions of the pattern periodicity \( e^a \) and \( e^b \) are orthogonal. \( (\phi^a, \phi^b)_{e(\alpha,\beta)} \) are the phase modulations of the reference and the current configurations, of the two perpendicular signals forming the checkerboard pattern [53], and finally “\( \cdot \)” is the 2D Euclidean dot product.

Denoting by \( q \) the displacement field that warps the reference image \( P \) to the current image \( F \), the conservation of the optical flow writes:

\[
\Gamma^q(x) = \Gamma(x + u(x)).
\]

After identification of the phase modulations of the “frng” functions, the above equation leads to, \( \forall \in [\alpha, \beta] \):

\[
2\pi k \cdot x + \phi^a(x) = 2\pi k_0 \cdot (x + u(x)) + \phi^b(x).
\]

Hence, since \((e^a, e^b)\) defines an orthonormal basis, the displacement satisfies:

\[
u(x) = \sum_{\alpha, \beta} \frac{\phi^a(x) - \phi^b(x) + u(x)}{2\|e\|} e^\alpha.
\]

Localized Spectrum Analysis (LSA) is one of the techniques developed to extract displacement fields from images of periodic patterns. It presents interesting properties in this context. First, this is a spectral approach that minimizes the optical residual in the Fourier domain, thus drastically reducing the computing cost of displacement calculations [43]. Second, it can easily process checkerboard images and other types of periodic patterns such as 2D grids.

The present paper aims to introduce a new software, OpenLSA. It is the culmination of developments over the last decade in collaboration between LORIA, (Université de Lorraine, Nancy, France) and the “Experimental Mechanics” team of the Institut Pascal research group, Clermont-Ferrand, France. LSA has been utilized in numerous literature.

For the experimental mechanics community, challenges addressed between LORIA, (Université de Lorraine, Nancy, France) and the “Experimental Mechanics” team of the Institut Pascal research group, Clermont-Ferrand, France. LSA has been utilized in numerous publications dealing with material and structure characterization [35,43–53]. However, only an early Matlab software toolbox [54] based on this technique is available, which justifies the present contribution developed with Python.

2. Software description

2.1. Properties of images of periodic patterns

The images considered here exhibit specific properties. They consist of multiple repetitions of an initial pattern, arranged with a given period along two perpendicular directions. An illustration is provided in Fig. 1. Fig. 1(a) depicts an imaged specimen surface, on which a checkerboard has been engraved following the procedure detailed in [47]. On a white background, the unit cell of the pattern consists here of a black shape of width about 30 [\( \mu m \)]. This pattern is then repeated along the perpendicular directions \((e^a, e^b)\) of the square with the period \( p = 30\sqrt{2} [\mu m]\), as illustrated by the close-up in Fig. 1(b).

The acquired image by the camera sensor can thus be modeled as the summation of two perpendicular 1D-periodic signals. In practice, the wave vector \( k \), defined for both \( \alpha \) and \( \beta \) directions, uniquely characterizes the direction and the period of such a signal. Specifically, its period corresponds to the inverse of the norm of the wave vector, i.e. \( p = 1/\|e\| \) and the repetition direction is given by the vector direction.
Applying 1D LSA to the 1D deformed pattern provides the raw phase modulation plotted in Fig. 2(b). However, the angle of a complex number is defined modulo 2\pi. Hence, two phenomena are induced.

- First, the angle of the WDFT is often spatially wrapped, so spatial unwrapping must be performed. This spatial unwrapping consists of adding pixel-wise integer multiple of 2\pi, so that the 2\pi-jumps of wrapped maps are removed. \( \psi_\ell (u_0) \) maps collect these pixel-wise integers, as illustrated in Fig. 2(c) for the considered 1D case.

- Second, the unwrapped phase maps are obtained modulo 2\pi, with the same multiple of 2\pi for all the pixels. A pairing is, therefore, performed between the reference and current phase modulations to obtain the value of this multiple. Plugging Eq. (6) into Eq. (3) gives, \( \forall \ell \in [\alpha, \beta] \):

\[
\hat{\kappa}_\ell \cdot u(x) = \frac{\psi_\ell^0(x) - \psi_\ell^0(x + u(x))}{2\pi} + c_\ell^0(x) + c_\ell^0(x + u(x))
\]

with \( \psi_\ell^0(x) = \text{angle} (F_{\omega_x}(\Gamma(x, \hat{\kappa}_\ell))) \) \( (7) \)

The pairing is calculated by assuming that the displacement \( u_0 \) at a given specific location \( x_0 \) is known. In practice, \( u_0 \) is obtained by tracking a marking defect. Indeed, Eq. (7) leads to

\[
c_\ell^0(x_0 + u_0) = c_\ell^0(x_0) + \frac{\psi_\ell^0(x_0) - \psi_\ell^0(x_0 + u_0)}{2\pi} - \hat{\kappa}_\ell \cdot u_0 \]

The bracket notation refers to the rounding function to the nearest integer. This rounding is required because of numerical errors. Considering the 1D case, the correction function is adjusted such that the displacement \( u_0 \), deduced from the phase modulation, equals 0 [px] at \( x_0 = 100 \) [px], i.e. at the center of the domain. This is illustrated in Fig. 2(d).

2.2.4. Code overview

To summarize, the displacement estimation from a pair of images by LSA consists of

(i) extracting the wrapped phase modulations from images \( I^0 \) and \( \Gamma \) using Eq. (6);

(ii) computing functions \( \langle c_\ell^f(x) \rangle_{\ell \in [\alpha, \beta]} \); spatially unwrapping all maps, and pairing the reference and current phase modulations based on the known displacement of a given point using Eq. (8);

(iii) solving Eq. (4) using a fixed-point algorithm, since the sought quantity \( u(x) \) is involved in both parts of the equality.

The organigram in Fig. 3 illustrates the code flow.

2.3. Software architecture

The software was designed to be straightforward for easy comprehension. The aim is to enable researchers, even those with little programming experience, to read, understand, and modify the code easily if needed.

The \( \text{lsa} \) class is the main class of this software. The \( \_\_\text{init}\_\_ \) method of this class initializes the problem. For any new case study, an image must be provided. The period and the orientation of the pattern, gathered in vector \( \vec{k} \), are determined by the \( \_\_\text{compute_vec_k} \) private method, which localizes the highest peak in the spectrum of the checkerboard image. The central peak is masked under the assumption that the pattern pitch is lower than max_pitch, which is set by default to 30 [pixels]. This corresponds to masking the central part of the spectrum: the pattern pitch is sought in the spectrum with a frequency larger than 1/max_pitch. Similarly, a lower bound is also introduced, defined by the variable min_pitch, which is set by default to 2\sqrt{2} [px]. For a checkerboard, the pattern pitch corresponds to the diagonal of a white or black square. This default value thus encodes each square by 2 x 2 pixels, which reaches the lowest limit to avoid aliasing effects [57]. Once the highest peak is located, \( \hat{\kappa}_\ell \) is defined. It is rotated by a multiple of \( \pi/2 \) to be as close as possible to the initial angle, which is set to 0 [rad] by default. The class constructor \( \_\_\text{init}\_\_ \) initializes \( \vec{k} \) if an image is given; otherwise, \( \vec{k} \) must be provided by the user. \( \vec{k} \) is then defined by rotating \( \vec{k} \) by \( \pi/2 \). \( \hat{\kappa}_\ell \) and \( \vec{k} \) define an attribute of the \text{OpenLSA} class, concatenated into the \text{vec_k} list.

After creating the class and defining the wave carriers, the window required for the WDFT needs to be elaborated. The \text{compute_kernel} method builds the Gaussian kernel with a standard deviation equal to \( \text{std} \). std corresponds to \( \sigma \) in the previous section. If \( \text{std} \) is not provided, it is defined by default as the smallest acceptable value, i.e., the period of the wave carriers [53].

At this stage, it is possible to compute the phase modulation for every image. This is the objective of the \text{compute_phases_mod} method. In practice, for each wave carrier direction \( \ell \in [\alpha, \beta] \), the method calculates the value of the angle taken by the WDFT of the checkerboard image at the corresponding frequency, cf Eq. (6). The modulus of the WDFT is also computed. It defines the region of interest roi if not provided. roi corresponds to the locations where the modulus of the WDFT is greater than an arbitrary value, set by default to a fifth of its maximum. The obtained phase modulations are subsequently unwrapped within the roi.

The \text{temporal_unwrap} method pairs phase modulations according to their corresponding images. To achieve this, a feature is extracted from the reference image, and its position defines a marker. This marker is then identified in the current image to facilitate pairing. The feature is usually a bright spot in a part of the pattern that exhibits correct encoding. Choosing a bright spot helps ensure that it is not caused by dust within the optical system. Since dust particles typically create black spots, the correct encoding of the pattern is determined by thresholding the modulus of the WDFT of the image at the pattern frequency by 75% of its maximum value. This ensures the selected
Fig. 2. Illustration of the spatial and temporal unwrapping necessary for the correct estimation of phase modulation. (a) 1D case: homogeneous stretching of a 1D sample. The displacement is affine, as is the associated phase modulation when considering a perfect pattern of $5 \, \text{px}$ period; (b) Raw wrapped phase modulation, obtained from the “angle” function applied to the WDFT of the stretched pattern; (c) Spatially unwrapped phase and its corresponding correction function; (d) Spatially and temporally unwrapped phase modulation, once the phase modulation of the stretched configuration has been paired to return the expected null displacement at the center of the specimen. The correction function is thus updated accordingly.

Fig. 3. Organigram of the code flow when OpenLSA is applied on an image pair $(\mathbb{I}^0, \mathbb{I})$ to retrieve the displacement $u$ that operates from the image $\mathbb{I}^0$ to the image $\mathbb{I}$.

Finally, the compute_displacement method solves Eq. (4) at each pixel of the region of interest. The displacement is defined as a complex number. The real (resp. imaginary) part corresponds to the displacement along the $e_1$ (resp. $e_2$) direction. A fixed-point algorithm is implemented for this purpose. The associated stopping criterion, defined by the average of the Euclidean norm of iterative corrections, is set by default to $5 \times 10^{-4}$. Moreover, the iterative process is stopped when the number of iterations reaches the value specified by max_iter, set to 15 by default. An option permits the initialization of this fixed point algorithm with the displacement returned by the DIS optical flow algorithm to reduce computing time. Moreover, when solving Eq. (4), the current phase modulation is interpolated at the location given by the iteratively-estimated displacement $u^{\text{est}}$, i.e. $(\varphi^T(x + u^{\text{est}}))_{(x \in [\alpha, \beta])}$. Since the region of interest is defined in the reference state, for pixels close to its boundary, the zero initial guess of the displacement, i.e. $u^{\text{init}} = 0$, might lead to interpolating the current phase at a point of the image which does not correspond anymore to the specimen surface. Initializing the compute_displacement method with the displacement returned by the DIS optical flow algorithm also prevents such an issue.
2.4. Software functionalities

The proposed code computes phase modulations of two orthogonal wave carriers within a pair of images. It deduces the displacement field from these phase modulations. The properties of the pattern (\(k_x\) and \(k_y\)) and the LSA parameters (wave vectors characterizing the pattern periodicity) are automatically computed. Two additional functionalities are introduced at this stage. They are discussed in the next two sections.

2.4.1. Phase and phases classes

The code introduces two additional classes related to the phase modulation quantity to facilitate its use. A phase modulation corresponds to the pixel-wise shift of an imaged pattern to its theoretical wave carrier. The Phase class encapsulates this information: its attributes include the wave carrier, defined by its wave vector denoted vec_\(k\), and the phase modulation, formatted as a Numpy array and called data. An additional attribute corresponds to the dimensions of the data array. Several methods are defined within the Phase class. For example, the unwrap and interp methods spatially unwrap and interpolate the phase maps, respectively. Additionally, the save and load methods aid in backing up results using the npz Numpy compressed file format.

The Phases class primarily consists of a list of objects defined as instances of the Phase class. Applying the main methods, such as unwrap and interp, of the Phase class to an object of the Phases class propagates them to each phase it encapsulates. Since images contain two phase modulations, one for each wave carrier, manipulating objects of the Phases class greatly enhances the ease of code reading.

2.4.2. Reducing sensor noise effect considering an image stack

An additional method called compute_refstate_from_im_stack is proposed for computing an averaged reference phase modulation from a stack of images. Indeed, it is often recommended to acquire several pictures of the unloaded reference state to minimize the impact of sensor noise on displacement estimation. Nevertheless, directly averaging the pictures may potentially result in a biased estimation of the noiseless reference phase map. Indeed, micro-vibrations (if any) occurring while capturing the stack of reference images can induce a bias in the phase map of the resulting averaged reference image. Here, the phase modulation is computed for each image of the stack, and the rigid body motion with respect to the first image of this stack is estimated and removed. A consistent average phase modulation is then computed.

The available images are also used to compute the noise floor level associated with the proposed measurement. For this purpose, the displacement and the strain fields that warp the first image to each other images in the stack are computed using the aforementioned phase modulations. This allows the estimation of the metrological performance of the measurement tool:

- The standard deviation of each component of the displacement and strain is calculated pixel-wise. The input variable display of the compute_refstate_from_im_stack method can be set to True for displaying such standard deviation maps. The global standard deviation of each component is also computed. It defines the measurement resolution [58]. These values are directly displayed in the terminal.
- The spatial resolution is also computed. The spatial resolution corresponds here to the smallest period of a sine displacement measured with a bias equal to \(\lambda\%\) or less. Here, \(\lambda = 10\%\) to be consistent with other studies dealing with the metrological performance of full-field measurement techniques [33,38,44]. The closed form expression of this quantity given in [59] is used here.
- The Metrological Efficiency Indicator (MEI) [60] is finally estimated for the displacement and the strain. For both quantities, only the component of the maximal measurement resolution is kept for calculating an upper bound.

Since the first image of the stack defines the reference coordinate system for the averaged phase modulations, it must also be used to initialize the problem for further LSA calculations. To avoid misuse, the compute_refstate_from_im_stack method runs the OpenLSA constructor with the first image and returns it, along with the calculated reference phase modulations and the kernel employed for it.

2.4.3. Access to image stored on an s3 server

In our research institute, data are stored in a lasting manner using an s3 server. For this purpose, OpenLSA code is written in such a way that access to an image stack is possible when an s3_dictionary is provided. This option is especially implemented for the OpenLSA constructor, which can thus build a reference phase from images stored in a single folder in such a s3 server.

3. Illustrative examples

3.1. Highlights of the main OpenLSA features

As a typical example, the OpenLSA software is applied here to images obtained during a compression test of a wood specimen. The images have already been used for another purpose in [52]. The specimen is made of fir, with dimensions of 50 × 35 × 15 [mm\(^3\)]. It is subjected to a load of 955 [N], see picture in Fig. 4(a). A checkerboard pattern was deposited on the observed surface of the specimen with the procedure detailed in [61]. It is worth noting that since then, the marking procedure has been simplified by using a laser engraver [47]. The deposited pattern features squares 100 [\(\mu\text{m}\)] wide. It is encoded with a 12-bit Sensicam CCD camera featuring 1376 × 1040 pixels, see the reference image \(I^0\) in Fig. 4(b). The close-up view in Fig. 4(c) displays the white and black squares.

First, the properties of the wave carriers are computed. For illustration, the Discrete Fourier Transform is applied to the image \(I^0\). The resulting spectrogram is depicted in Fig. 4(d). According to the details provided in Section 2, a portion of this representation is masked to estimate a preliminary wave vector. This mask is displayed with a reduced transparency in Fig. 4(d). Both the \(k_x\) and \(k_y\) wave vectors are plotted in Fig. 4(e), which is a close-up view of the spectrogram of image \(I^0\).

In this example, the standard deviation of the Gaussian window used in LSA is set to its minimum value, i.e., the pattern period. This length corresponds here to the minimum distance between two black or white squares obtained on the diagonal of the squares. It is then possible to extract the phase modulations from the reference image along both the \(k_x\) and \(k_y\) directions at the frequency of \(|\|k\||\). They are illustrated in Figs. 5(a) and 5(b). These figures also show the automatic selection of the region of interest, which considers all the pixels for which the modulus is greater than 25% of its maximum value. The pixels not included within this region of interest are not displayed. If any, they appear in white in Figs. 5(a) and 5(b) and in all subsequent figures.

As discussed in Section 2, the raw estimation of the phase maps is computed pixelwise and modulo 2\(\pi\). The resulting raw phase maps are unwrapped, and the results are shown in Figs. 5(c) and 5(d). The same procedure is then applied to the current image \(I\). The corresponding unwrapped phases are shown in Figs. 5(e) and 5(f). The temporal pairing with the reference image \(I^0\) has been achieved through the calculation of the movement that followed an arbitrary point \(x_0\) located at the coordinates \([867, 412]\), as indicated by a red cross in Fig. 4(b). This point corresponds to a feature, illustrated in Fig. 6, easily trackable from one image to another.

Finally, the displacement maps along directions 1 and 2 are deduced from the two phase modulations, according to Eq. (4). They are plotted in Fig. 7(e).
The strain maps are displayed in Fig. 7(e). The following remarks can be drawn from this Figure:

- As expected with this kind of test, the order of magnitude of the strain is quite small. Indeed, the longitudinal strain $\varepsilon_{22}$ remains below 1%, in absolute value (see Fig. 7(d)). In other words, the difference in displacement is about 1 pixel between the top and bottom of the specimen.
- $\varepsilon_{11}$ of Fig. 7(c) illustrates the Poisson effect. With a mean value of about $2.4 \times 10^{-4}$ [–], $\varepsilon_{11}$ reveals the effect of sensor noise propagation on the strain maps. Interestingly, the strain noise observed here does not correlate with the pattern, contrary to its counterpart observed with DIC [35,36] in similar cases.
- Because it processes optimized patterns, LSA provides displacement/strain maps with the highest possible metrological performance. This is highlighted by the ability of the method to illustrate material heterogeneity within strain maps. Here, early and late woods are identifiable in the strain maps. This is particularly true for the normal strain $\varepsilon_{22}$, and the shear strain $\varepsilon_{12}$; see Figs. 7(c) and 7(e).

3.2. Using an image stack of the reference state for improving metrological performance

The second example illustrates the ability of the OpenLSA program to take advantage of an image stack of the reference state to improve the metrological performance of the measurement. For this purpose, a set of images already used in [62] and available here [63] are employed. These images correspond to a tensile test of a wood specimen and come from Test #2 of [62]. This test has been selected here because the observed strain maps are of low level and consequently show a relatively poor signal-to-noise ratio. A checkerboard pattern has been deposited onto the wood surface following the procedure presented in [47]. An interested reader will find more details about the test in [62].

Images have been duplicated onto our university’s s3 server to facilitate code testing, example manipulation, and the illustration of the s3 access code option. Moreover, images are cropped to 1024 x 1024 pixels to reduce the computing time of this illustration. The crop location includes a few tree rings, which induce strain map heterogeneity to emphasize the gain of using an image stack. Two sets of images are available:
Fig. 5. Phase maps associated with images \( I^0 \) and \( I^1 \) (a) and (b) are the raw phase maps associated with image \( I^0 \) along the wave carrier directions: (a) (resp. (b)) corresponds to the carrier of wave vector \( k_\alpha \) (resp. \( k_\beta \)). The results are wrapped because they are given modulo \( 2\pi \). (c) and (d) show the same phase maps after unwrapping. (e) and (f) display the unwrapped phase maps associated with image \( I^1 \).

Fig. 6. Close-up of the green square in Fig. 4(b), which is centered on a specific pattern feature. The later is tracked across images to solve the phase modulations’ temporal pairing.

- the first set is called \( I^0 \) and contains 201 images corresponding to the reference state (loading \( F = 0 \) [N]): \( I^0 = (I_0^0)_{0 \leq i \leq 200} \);
- the second set, called \( I^1 \), was exceptionally taken to capture the current state (loading \( F = 3200 \) [N]). It collects 201 images: \( I^1 = (I_0^1)_{0 \leq i \leq 200} \).

Three image procedures are applied to these sets.

- First, LSA is applied using only the first image of each set. Let \( u_{\text{im}}^1 \) be the retrieved displacement. This displacement corresponds to the regular use of LSA, which relies on a pair of images.
- LSA is then applied using the whole set \( I^0 \) and the first image of the \( I^1 \) set. The obtained displacement is denoted by \( u^\text{im stack} \). It corresponds to the proposed strategy, in which an image stack of the reference state is taken to improve the metrological performance, and one current image is indeed used to describe the current state. Processing the \( I^0 \) image set, which corresponds to multiple views of the same material state, allows the calculation of the metrological performances in this case. The values are given in Table 1.
- Finally, the unusual fact that an image stack of the current state is also available makes the calculation of a noise-reduced current
Fig. 7. (a) and (b) are respectively the $\varepsilon_1$ and $\varepsilon_2$ components of the displacement. (c), (d), and (e) are the components of the strain tensor in the $(\varepsilon_1,\varepsilon_2)$ plane. $\varepsilon_{22}$ corresponds to the longitudinal strain along direction 2. This is a compression test, so $\varepsilon_{22}$ is negative.

The focus is made here on the strain maps, on which the effect of noise is clearly noticeable. Fig. 8 gathers the results. Fig. 8(a) shows $\varepsilon_{22}^{\text{im}} = \varepsilon_{22}(\varepsilon_1^{\text{im}})$ maps. On top of the tree rings, the usual peach-skin aspect of the noise is visible. The latter is indeed perceptible for strain values of a few $10^{-3}$. The standard deviation of the error between the 22-strain maps elaborated from $\varepsilon_1^{\text{im}}$ and $\varepsilon_n^{\text{nr}}$ is equal to $8.78 \times 10^{-4}$ [px].

Fig. 8(b) shows $\varepsilon_{22}^{\text{im stack}} = \varepsilon_{22}(\varepsilon_1^{\text{im stack}})$. Tree rings are also visible, and the map is slightly less noisy. The standard deviation of the error between the 22-strain maps elaborated from $\varepsilon_1^{\text{im stack}}$ and $\varepsilon_n^{\text{nr}}$ is equal to $6.06 \times 10^{-4}$ [px].

By definition, the variance of the strain maps is proportional to the average of the variances of the phases of the reference and the current states [59]. When a large image stack is considered for estimating the reference phase, the latter can be considered as noiseless. Consequently, the variance of the strain maps should be divided by 2, and thus its standard deviation by $\sqrt{2}$. This property is almost verified here:

$$\frac{\text{std}(\varepsilon_{22}^{\text{im}} - \varepsilon_{22}^{\text{nl}})}{\text{std}(\varepsilon_{22}^{\text{im stack}} - \varepsilon_{22}^{\text{nl}})} = \frac{8.78 \times 10^{-4}}{6.06 \times 10^{-4}} = 1.45 \approx \sqrt{2}.\quad (9)$$

Finally, Fig. 8(c) plots the $\varepsilon_{22}^{\text{im}}$ and $\varepsilon_{22}^{\text{im stack}}$ strains along the line shown in Fig. 8(a) (black line) and Fig. 8(b) (red line), respectively.
The dashed blue line corresponds to \( \epsilon_{22}^{\text{nr}} \) at the same location. Again, \( \epsilon_{22}^{\text{im stack}} \) is closer to \( \epsilon_{22}^{\text{nr}} \). This is confirmed by the standard deviation of the difference between \( \epsilon_{22}^{\text{im stack}} \) and \( \epsilon_{22}^{\text{nr}} \) along this line:

\[
\text{std}(\epsilon_{22}^{\text{im stack}} - \epsilon_{22}^{\text{nr}}) = 8.86 \times 10^{-4} > 5.21 \times 10^{-4} = \text{std}(\epsilon_{22}^{\text{im stack}} - \epsilon_{22}^{\text{nl}}) \tag{10}
\]

4. Impact

Material comprehension and optimization are significant motivations for many studies in the experimental mechanics community. Undoubtedly, full-field measurement techniques have revolutionized how experimentalists perceive their experiments. Material testing is currently undergoing a breakthrough, with experimentation designed to leverage the rich insights provided by full-field measurements \([64, 65]\). Thanks to its availability as commercial or free software, Digital Image Correlation (DIC) is routinely used for computing displacement and strain fields. However, DIC has drawbacks, such as its inability to deal with patterns optimized for metrological performance. Localized Spectrum Analysis (LSA), the backbone of OpenLSA, is specifically designed to process such optimal patterns, such as checkerboards. While an early in-house Matlab software toolbox is available \([54]\), by publishing our Python code, we aim to assist DIC users and experimentalists in exploring the alternative offered by the LSA method. The outstanding metrological performance of LSA \([43]\) ensures its valuable use.

The deposition of patterns was previously a challenge that limited the adoption of LSA. However, this limitation is now outdated, thanks to the ease of use of a laser engraver \([47]\), and more recently, the ability to deposit checkerboard patterns using similar materials and techniques as those used for strain gauges \([62]\).

We aim to contribute to the broader scientific community by sharing our methods, techniques, and findings, in the hope that they will assist other researchers in understanding and potentially building upon our work.

Moreover, LSA has facilitated collaborations for us, both at a national level \([51,66]\) and internationally \([46,49,50,67]\). Making this code accessible will contribute to the dissemination of these relevant full-field measurement techniques in the experimental mechanics community, serving as a valuable resource for experimentalists tackling similar problems or building upon related concepts. Additionally, the software code is intended for educational purposes.

Making our research publicly available also enhances its reproducibility. Other researchers can now utilize and validate our methods, leading to more robust and reliable scientific results.

Finally, this work contributes to the ongoing efforts aimed at standardizing full-field measurement techniques, particularly in surface marking. The geometry of a periodic pattern, such as a checkerboard, can be easily defined with a limited number of parameters, making it conducive to standardization.

5. Conclusions

In this article, we introduce OpenLSA, a software that applies Localized Spectrum Analysis (LSA) to images of periodic patterns to retrieve the displacement that warps them. LSA demands low computations. It processes optimal patterns, such as checkerboard patterns, and provides full-field measurements with the best metrological performance. Dedicated to the experimental mechanics community, this program will significantly facilitate the adoption of LSA. Written as a Python class, its use is easy, with many automated steps. Thanks to its open-source nature, it encourages community collaboration.

Further work will propose integrating the dedicated deconvolution algorithm for LSA, which will enhance the metrological performance of the measuring tool. Lastly, an additional extension that would enhance the framework’s usability is the integration of a camera model to take it into account within full-field estimation.

CRediT authorship contribution statement

Benoît Blaysat: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization. Frédéric Sur: Writing – review & editing, Writing – original draft, Software, Methodology, Formal analysis, Conceptualization. Thomas Jailin: Writing – review & editing, Writing – original draft, Software, Methodology. Adrien Vinel: Writing – review & editing, Writing – original draft, Software, Methodology. Michel Grédiac: Writing – review & editing, Writing – original draft, Methodology, Formal analysis, Conceptualization.
Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: BENOIT BLAYSAT reports financial support was provided by Université Clermont Auvergne. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data is linked (https://github.com/BenoitBlaysat/OpenLSA.git).

Acknowledgments

The authors acknowledge support from the French government research program “Investissements d’Avenir” through the IDEX-ISITE initiative 16-IDEX-0001 (CAP 20-25) and the AURA regional council.

References

Benoît Blaysat et al.


JCGM Member Organizations. International vocabulary of metrology – Basic and general concepts and associated terms (VIM), vol. 200. BIPM; 2012.


JCGM Member Organizations. International vocabulary of metrology – Basic and general concepts and associated terms (VIM), vol. 200. BIPM; 2012.


