Journal of Mathematical Imaging and Vision (2018) 60:634–650
https://doi.org/10.1007/s10851-017-0779-4

Rendering Deformed Speckle Images with a Boolean Model

Frédéric Sur[1] · Benoît Blaysat[2] · Michel Grédiac[2]

Received: 9 February 2017 / Accepted: 6 December 2017 / Published online: 14 December 2017

Abstract

Rendering speckle images affected by a given deformation field is of primary importance to assess the metrological performance of displacement measurement methods used in experimental mechanics and based on digital image correlation (DIC). This article describes how to render deformed speckle images with a classic model of stochastic geometry, the Boolean model. The advantage of the proposed approach is that it does not depend on any interpolation scheme likely to bias the assessment process, and that it allows the user to render speckle images deformed with any deformation field given by an analytic formula. The proposed algorithm mimics the imaging chain of a digital camera, and its parameters are carefully discussed. A MATLAB software implementation and synthetic ground-truth datasets for assessing DIC software programs are publicly available.

Keywords Boolean model · Random speckle rendering · Digital image correlation (DIC)

1 Introduction

In experimental mechanics, numerous variants of digital image correlation (DIC) are used to measure displacement or strain fields on a specimen subjected to a mechanical load and consequently deformed [50]. DIC methods are based on two images $\mathcal{I}$ and $\mathcal{I}'$ of the surface of the specimen, taken before and after deformation, respectively. The aim is to retrieve the mapping $\boldsymbol{\psi} : \mathbb{R}^2 \to \mathbb{R}^2$ which registers $\mathcal{I}'$ on $\mathcal{I}$ by maximizing a correlation or minimizing an optical residual between patches extracted from $\mathcal{I}$ and $\mathcal{I}' \circ \boldsymbol{\psi}$, where $\circ$ denotes the composition operator. With these notations, $\mathbf{u}$ such that $\boldsymbol{\psi}(\boldsymbol{\xi}) = \boldsymbol{\xi} + \mathbf{u}(\boldsymbol{\xi})$ is the so-called direct displacement field, $\mathbf{U}$ such that $\boldsymbol{\psi}^{-1}(\boldsymbol{\xi}) = \boldsymbol{\xi} + \mathbf{U}(\boldsymbol{\xi})$ being the inverse displacement field. In order to ensure that the image patches contain contrasted distinctive shapes, ink is often sprayed on the surface of the specimen, yielding a speckle pattern on the surface of the specimen. Figure 1 shows typical examples of speckle patterns.

While the amplitude of the displacement field can be as large as several tens of pixels across the image, the strain components, which are deduced from the gradient of the displacement field, are generally much lower than 1% in most structures made of engineering materials such as steel, wood, or concrete working in their normal operating range. Amplitudes greater than several percents are reached only in components made of soft materials such as elastomers for instance, or made of the previous materials when they are close to failure because of excessive load. Such small values make it necessary to carefully design estimation methods. Benchmarking the performance of DIC-based methods requires to compare the estimated displacement and the actual, yet unknown displacement. Ground-truth databases are thus needed, that is, pairs of synthetic speckle images, taken in a reference state on the one hand, and after deforming the image by a given prescribed mapping $\boldsymbol{\phi}$ on the other hand. It is worth noting that the displacement field returned by any DIC software program from such pairs of images is impacted by both the errors due to the rendering of the deformed artificial speckles, and the errors caused by the DIC program itself, while only the last ones are generally the quantity to be characterized. It is, therefore, of prime importance to have at disposal artifact-free artificial deformed speckles to be certain that only the errors due to the DIC programs are really estimated. There is indeed an active community

✉ Frédéric Sur
frederic.sur@loria.fr

Benoît Blaysat
benoit.blaysat@uca.fr

Michel Grédiac
michel.grediac@uca.fr

[1] Université de Lorraine. LORIA UMR CNRS 7503, INRIA projet Magrit, Campus Scientifique, BP 239, F-54506 Vandœuvre-lés-Nancy Cedex, France

[2] Institut Pascal-SIGMA, Campus Universitaire des Cézeaux, 27 rue Roche Genes, CS 20265, 63175 Aubière Cedex, France

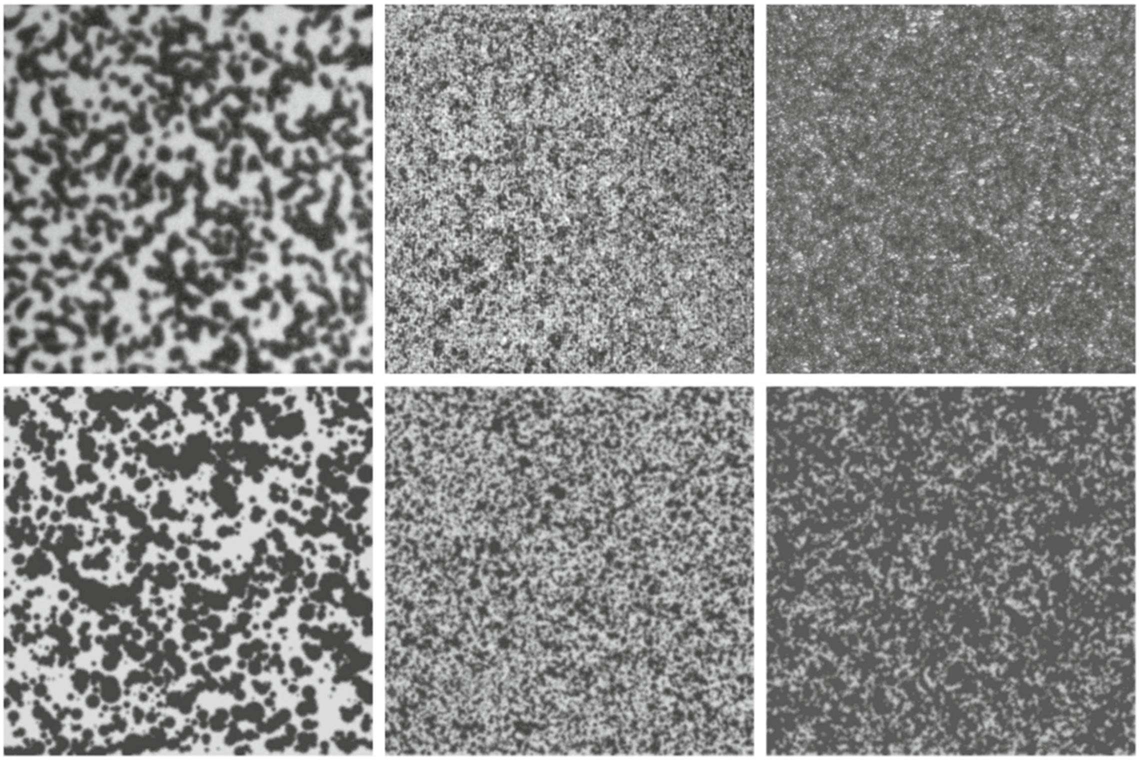


Fig. 1 First row: typical speckle images proposed in [47] for DIC program assessment. From left to right: close-up views of images from samples Sample3, Sample12, Sample13a from [47]. Second row: images rendered with the proposed algorithm. To generate these 300 × 300 images, $\sigma = 1$ pixel and speckle pattern parameters (same notations as in Table 2) are, from left to right: $\lambda = 2000$, $\mu = 3$ pixels (Poisson distribution of disk radii), $\gamma = 0.6$; $\lambda = 50{,}000$, $\mu = 0.5$ pixels (exponential distribution), $\gamma = 0.6$; $\lambda = 50{,}000$, $\mu = 0.7$ pixels (exponential distribution), $\gamma = 0.5$

concerned by the assessment of the metrological performance of DIC programs, as illustrated by numerous recent papers on this problem [2,7,8,11,13,16,18,26,27,38,41,52] for instance. The Society for Experimental Mechanics (SEM) proposes such ground-truth databases within the framework of the DIC challenge [47], providing the researchers with synthetic speckle images rendered by several methods of the literature, briefly reviewed below.

1.1 Synthetic Speckle Rendering for DIC

In [26,38,41,51,52], a method is presented to render a speckle image under a prescribed translation via phase modulation in Fourier domain. Translating images is, however, too limited to fully analyze the performance of DIC programs and more complicated displacement fields should be permitted. In [43], Fourier phase modulation is also used for interpolation with a prescribed quadratic displacement. As noted by the authors of [14], such a method should not introduce any bias provided that the underlying infinite resolution speckle image satisfies the Shannon–Nyquist condition, which is a quite strong assumption in practice. In addition to these methods, any prescribed displacement can be used by numerically interpolating a real or synthetic speckle image [4,16,19,28,51]. DIC assessment is, however, likely to be affected by the interpolation scheme. In order to reduce the influence of interpolation, a possibility is to generate low-resolution images by binning pixels of an interpolated high-resolution deformed speckle image [4,6,7,10,18,31,41].

TexGen [37] is a software program that generates synthetic speckle images based on Perlin noise [39]. Although the description of TexGen misses some details of the method, Perlin noise is intrinsically a discrete scheme: it is generated in [37] through a “super-sampled” image which gives the final image after a Monte Carlo integration mimicking the effect of the fill-factor of CCD sensors. This method is used for example in several assessment papers [11,12,14]. Alternatively, in the context of color images, it is proposed in a recent paper [5] to generate synthetic speckles as the sum of randomly distributed bell-shaped functions. Afterward, deformed speckles seem to be numerically integrated over a pixel with a quadrature rule. Note that this is similar to the filtered Poisson process described in [49].

The authors of [12,14] note that the algorithms used to render synthetic images are critical. They suggest that the interpolation schemes used in these algorithms may introduce a bias in the quantification of measurement uncertainties, which seems to make TexGen the method of choice among the available ones.

1.2 Contribution

The contribution of the present paper is to describe a novel method to render synthetic speckle images under any arbitrary displacement function. The advantage of the proposed approach is that it does not require any interpolation scheme. As well as TexGen [37], all what is needed is a closed-form expression for $\boldsymbol{\psi}$ or $\boldsymbol{\psi}^{-1}$ defined in the introduction. In particular, discontinuous displacement fields can be designed. Contrary to TexGen, the proposed method is based on an infinite resolution view of the speckle pattern lying on the specimen surface. Speckle patterns are modeled by a so-called Boolean model, and the rendering scheme rigorously follows the image acquisition chain in a digital camera, from the infinite resolution view to the digital image. The numerical accuracy of the method is carefully discussed to ensure that no bias is introduced when rendering the deformed images, to be certain that further assessments of DIC programs relying on such images return displacement fields free from any disturbance coming from the rendering stage. The ability of the proposed algorithm to produce real-looking patterns is illustrated in Fig. 1.

Section 2 is a reminder on the Boolean model, a classic model of stochastic geometry. This model gives an infinite resolution view of speckle patterns. Section 3 recalls the basis of image formation in a digital sensor and explains how to build up a digital image from this infinite resolution view, based on a Monte Carlo integration. The influence of algorithmic parameters is carefully questioned and a numerical assessment of the benefit of the proposed method over the popular rendering method based on interpolation is also provided to the reader. A short discussion on realistic noise rendering is also proposed. Section 4 concludes the discussion.

In order to allow the researchers in the field of experimental mechanics to use the rendered speckle images as ground-truth databases, we make a MATLAB code and several databases publicly available on the following webpage: http://members.loria.fr/FSur/software/BSpeckleRender/. Scripts permitting to reproduce the experiments of the present paper are also available.

1.3 Notations

Boldface letters denote vectorial quantities. Roman alphabet is used for sampled pixel coordinates (integer values) and Greek alphabet for pixel coordinates in the infinite resolution image (real values). The indicator function of any planar set $\mathcal{S}$ is denoted by $\mathbb{1}_{\mathcal{S}}$ (that is, for any $\boldsymbol{\xi} \in \mathbb{R}^2$, $\mathbb{1}_{\mathcal{S}}(\boldsymbol{\xi}) = 1$ if $\boldsymbol{\xi} \in \mathcal{S}$, and $\mathbb{1}_{\mathcal{S}}(\boldsymbol{\xi}) = 0$ otherwise). Provided that these quantities exist, $\mathcal{A}(\mathcal{S})$ and $\mathcal{P}(\mathcal{S})$ denote the area and the perimeter of $\mathcal{S}$, respectively.

2 Modeling Speckle Patterns

Since a speckle image is generally obtained by spraying ink onto the surface of the tested specimen, we propose to model speckle images as the superposition of random black shapes on a white background. On the non-deformed surface, these shapes are simply disks.

2.1 Non-deformed Speckle Images

The proposed approach consists in modeling speckle patterns within the framework of Boolean models. It is based on the notion of (homogeneous) Poisson point process of intensity λ on a bounded domain $\Omega \subset \mathbb{R}^2$, which comes down to drawing a random variable N with a Poisson distribution of mean $\lambda\mathcal{A}(\Omega)$, and N independent points, uniformly distributed in Ω [3]. Such a Poisson point process $\{\mathbf{d}_i, 1 \le i \le N\}$ and an independent sequence of independent and identically distributed compact sets $\{\mathbf{D}_i, 1 \le i \le N\}$ being given, a random set $\mathbf{P}$ is defined as the union of all $\mathbf{D}_i$ translated by $\mathbf{d}_i$:

$$\mathbf{P} = \cup_{1 \le i \le N} (\mathbf{d}_i + \mathbf{D}_i)\,. \tag{1}$$

The random set $\mathbf{P} \subset \mathbb{R}^2$ follows a so-called Boolean model, a classical model of stochastic geometry [42,44,48] that has been applied in various situations such as, for instance, material analysis [45], distribution of trees in a forest [22], or film grain modeling in a resolution-free way [34] (see also [35] and the software implementation [36]). We shall come back to this latter paper in the discussion of the proposed algorithm.

The infinite resolution image of the speckle pattern is defined as the following binary image:

$$\mathcal{I}_0(\boldsymbol{\xi}) = \mathbb{1}_{\mathbf{P}}(\boldsymbol{\xi}), \tag{2}$$

value 0 corresponding to the bright background and value 1 to the dark ink. Since speckle is obtained by imaging the surface of the specimen sprayed by ink, each speckle pattern $\mathbf{D}_i$, corresponding to an ink droplet, is modeled as a disk of radius R_i centered at the origin. We model the R_i as independent random variables, identically distributed following an exponential distribution of mean μ (alternative choices are discussed in Sect. 2.3).

To sum up the discussion, it can be said that simulating the Boolean model of Eq. (1) or, equivalently, the infinite resolution speckle image of Eq. (2), is achieved through the following steps:

(1) Generate a realization of the homogeneous Poisson process giving the $\mathbf{d}_i$ by drawing a Poisson count N (intensity λ) and generate N points $\mathbf{d}_i$ uniformly distributed in the continuous image domain Ω;
(2) Draw N disks $\mathbf{D}_i$, each radius being the realization of an exponential random variable of mean μ, centered at the $\mathbf{d}_i$.

2.2 Deformed Speckle Images

Let $\boldsymbol{\phi}$ be the mapping giving the coordinates $\boldsymbol{\phi}(\boldsymbol{\xi})$ in the non-deformed image of a material point of coordinates $\boldsymbol{\xi}$ in the deformed image. The infinite resolution deformed speckle image $\mathcal{I}_0'$ is still a binary image, defined by

$$\mathcal{I}_0'(\boldsymbol{\xi}) = \mathcal{I}_0(\boldsymbol{\phi}(\boldsymbol{\xi})) = \mathcal{I}_0 \circ \boldsymbol{\phi}(\boldsymbol{\xi}). \tag{3}$$

The mapping $\boldsymbol{\psi}$, introduced in Sect. 1, which DIC aims at retrieving, satisfies $\boldsymbol{\psi} = \boldsymbol{\phi}^{-1}$. Prescribing $\boldsymbol{\phi}$ is equivalent to prescribing the inverse displacement field $\mathbf{U}$. As we shall see in Sect. 3, the proposed framework permits us to render finite resolution deformed speckle images for any mapping $\boldsymbol{\phi}$ (thus any inverse displacement field $\mathbf{U}$) defined for any $\boldsymbol{\xi} \in \Omega$. Not only are the positions of the disks mapped through $\boldsymbol{\phi}$, but also the shape of these disks.

However, DIC-based methods give an estimation of $\boldsymbol{\psi}$ (or, equivalently, of the direct displacement field $\mathbf{u}$). This necessitates to assess the metrological performance of a DIC-based method by comparing its output with the *inverse* of the prescribed $\boldsymbol{\phi}$. Direct and inverse displacement fields are linked by $\mathbf{U}(\mathbf{x}+\mathbf{u}(\mathbf{x})) = -\mathbf{u}(\mathbf{x})$ but it is, in general, not easy to obtain a closed-form expression of $\mathbf{u}$ from $\mathbf{U}$. Except for simple cases (for instance, if the prescribed mapping $\boldsymbol{\phi}$ is a uniform translation by a vector $\mathbf{a}$, $\boldsymbol{\psi}$ is a translation by $-\mathbf{a}$), calculating $\boldsymbol{\phi}^{-1}$ would require numerical methods such as fixed-point iteration, which potentially give additional biases when assessing the performance of DIC. For this reason, when $\boldsymbol{\phi}$ is not explicitly invertible, we suggest to consider $\mathcal{I}_0'$ as the non-deformed image and $\mathcal{I}_0$ as the deformed one, the mapping given by DIC being in this case directly comparable to the prescribed $\boldsymbol{\phi}$. This must be explored by further studies beyond the scope of this paper.

Since we do not address DIC assessment in the present paper, we keep on calling $\mathcal{I}_0$ the reference, non-deformed speckle image, and $\mathcal{I}_0'$ the deformed image.

2.3 Two Properties of Boolean Models

We finish the discussion of Boolean models by two properties which give an insight into the role of the λ and μ parameters. Both quantities will also turn out to be useful in the following sections. The infinite resolution image $\mathcal{I}_0$ takes either 0 or 1 values. It is known [42,48] that the proportion of 1-values (corresponding to a covering ratio of the random set $\mathbf{P}$ over the resulting image supported by Ω), is given by

$$\kappa(\mathbf{P} \cap \Omega) = 1 - \exp\left(-\lambda E\left(\mathcal{A}(\mathbf{D}_0)\right)\right) \tag{4}$$

where $E(\mathcal{A}(\mathbf{D}_0))$ is the expectation of the area of the identically distributed random sets $\mathbf{D}_i$.

It is also known (see, e.g., [42,48] for convex shapes $\mathbf{D}_i$, and [20] without the convexity assumption) that the perimeter of $\mathbf{P}$ over the resulting infinite resolution image is given by

$$\theta(\mathbf{P} \cap \Omega) = \lambda \mathcal{A}(\Omega) E\left(\mathcal{P}(\mathbf{D}_0)\right) \exp\left(-\lambda E\left(\mathcal{A}(\mathbf{D}_0)\right)\right) \tag{5}$$

where $E(\mathcal{P}(\mathbf{D}_0))$ is the expectation of the perimeter of the random $\mathbf{D}_i$.

Here $E(\mathcal{A}(\mathbf{D}_0)) = \pi E(R^2)$ and $E(\mathcal{P}(\mathbf{D}_0)) = 2\pi E(R)$ since the random sets are disks of random radius R. Table 1 gives the expression of several moments of R, of the covering ratio, and of the perimeter for four potential probability distribution of $R \geq 0$.

In the experiments of the present paper, we use an exponential distribution of mean μ. Consequently, the covering ratio is given by:

$$\kappa = 1 - \exp(-2\pi\lambda\mu^2) \tag{6}$$

Table 1 Distribution of the random radii $R \geq 0$ of the disks in the Boolean model

Distribution	$E(R)$	$\mathrm{var}(R)$	$E(R^2)$	Covering κ (Eq. (6))	Perimeter θ (Eq. (7))
Exponential	μ	μ^2	$2\mu^2$	$1 - e^{-2\pi\lambda\mu^2}$	$2\pi\lambda\mathcal{A}(\Omega)\mu e^{-2\pi\lambda\mu^2}$
Uniform on $[0, 2\mu]$	μ	$\mu^2/3$	$4\mu^2/3$	$1 - e^{-4\pi\lambda\mu^2/3}$	$2\pi\lambda\mathcal{A}(\Omega)\mu e^{-4\pi\lambda\mu^2/3}$
Poisson	μ	μ	$\mu + \mu^2$	$1 - e^{-2\pi\lambda(\mu+\mu^2)}$	$2\pi\lambda\mathcal{A}(\Omega)\mu e^{-2\pi\lambda(\mu+\mu^2)}$
Log-normal	μ	σ_R^2	$\sigma_R^2 + \mu^2$	$1 - e^{-\pi\lambda(\sigma_R^2+\mu^2)}$	$2\pi\lambda\mathcal{A}(\Omega)\mu e^{-\pi\lambda(\sigma_R^2+\mu^2)}$

In the case of the log-normal law (used in the Boolean model of [34]), R is such that $\log(R)$ follows a Gaussian distribution of mean $\mu' = \log(\mu^2/\sqrt{\mu^2 + \sigma_R^2})$ and variance $\sigma'^2 = \log(1 + \sigma_R^2/\mu^2)$

and the perimeter by:

$$\theta = 2\pi\lambda A\mu \exp(-2\pi\lambda\mu^2). \tag{7}$$

As expected, the covering ratio being fixed, the larger the mean number of disks λ (with an average radius μ^2 inversely proportional to λ), the larger the perimeter κ.

The problem of determining the best parameters (intensity, distribution of the random shapes) to design a Boolean model that mimic a real image is the subject of a quite large literature, see for instance the references in [42,48]. Although comparing the examples shown in Fig. 1 shows that the proposed rendering method provides speckle images which are quite similar to those currently used for DIC program assessment, it could be interesting to address such issues in a separate study in order to design Boolean models for real speckle images used in experimental mechanics.

3 Digital Speckle Image Rendering

In this section, we explain how a finite resolution digital image is rendered from the infinite resolution speckle image defined in Sect. 2.

3.1 Acquisition of an Image Through a Digital Sensor

Any digital image is made of a collection of sampled pixels whose discrete gray value is the result of the integration of photons across the optical device. More precisely, if $\mathcal{I}_0$ is an infinite resolution view of a planar surface of a specimen, the digital image $\mathcal{I}$ is given by the following equation:

$$\mathcal{I} = Q\, S\, G\, H\, \mathcal{I}_0. \tag{8}$$

In this classical setting (illustrated by example in Figure 3 of [32]), Q quantizes any real number over b bits (thus 2^b possible values), S is the sampling operator on the pixel grid, G represents the optical distortions (such as radial distortions, or convolution with the point spread function—PSF—of the optical device), and H is the geometric transform which maps the planar image $\mathcal{I}_0$ to the image plane. In the pinhole camera model, H is a homography [24]. In this framework, further processing steps such as γ-correction, JPEG compression, or demosaicing (discussed in [5,9]) are not taken into account. The digital image $\mathcal{I}$ corresponds to the output of a linear digital camera such as a specialized scientific camera or the raw output of any consumer camera. Digital noise, a phenomenon which is also overlooked by the model of Eq. (8), is discussed in Sect. 3.5.

We simplify this generic model by assuming that the infinite resolution view is parallel to the image plane. Homography H thus corresponds to a scale change and a rotation; here H is the identity transform without loss of generality. Moreover, optical distortions are restricted to the convolution with the PSF, assumed here to be an isotropic 2D Gaussian function of standard deviation σ, denoted by G_σ. We also bring back the contrast of the imaged speckle to a $[0, \gamma]$ range by multiplication by a contrast parameter $0 < \gamma \leq 1$.

The digital image $\mathcal{I}$ eventually writes:

$$\mathcal{I} = Q\, S\, (\gamma\, G_\sigma * \mathcal{I}_0) \tag{9}$$

where $*$ denotes the convolution between two 2D functions.

The Fourier transform of the Gaussian function G_σ being $\mathcal{F}(G_\sigma)(u, v) = e^{-2\pi^2\sigma^2(u^2+v^2)} = K\, G_{1/(2\pi\sigma)}(u, v)$ (K being some multiplicative constant), any signal component whose frequency is above, say, twice the standard deviation of $G_{1/(2\pi\sigma)}$ is smoothed out in the output image. Since Nyquist–Shannon sampling condition is satisfied as soon as the largest frequency component of the imaged pattern is smaller than half the sampling rate, this yields $2/(2\pi\sigma) < 1/2$ here, thus $\sigma > 0.64$ pixel. As a rule of thumb, we can say that no aliasing is present in the digital image $\mathcal{I}$ as soon as $\sigma > 1$ pixel, whatever the frequency components of the infinite resolution image $\mathcal{I}_0$. Of course, the practical effect of aliasing depends on the spectrum of the infinite resolution image. Indeed, although the “twice the standard deviation” rule is questionable, this spectrum most likely does not show any meaningful component above $1/(\pi\sigma)$. In practice with the Boolean images, it is probably sufficient to set $\sigma > 0.5$. However, it could be useful to deliberately generate aliased images by using very small σ, in order to assess the effect of aliasing on DIC algorithms.

To sum up, with Eq. (9), the gray level at any (discrete) pixel $\mathbf{x}$ of the digital image $\mathcal{I}(\mathbf{x})$ is given by

$$\mathcal{I}(\mathbf{x}) = Q\ (\gamma\, I(\mathbf{x})) \tag{10}$$

where $Q(t)$ is a quantized representation of any real value $t \in [0, 1]$ over b bits (typical values are $b = 8$, 12, 14, or 16 bits in high-end modern cameras), and $I(\mathbf{x})$ is given by the following integral:

$$\begin{aligned} I(\mathbf{x}) &= \iint G_\sigma(\mathbf{x} - \boldsymbol{\xi})\, \widetilde{\mathcal{I}}_0(\boldsymbol{\xi})\, \mathrm{d}\boldsymbol{\xi} \\ &= \iint G_\sigma(\boldsymbol{\xi})\, \widetilde{\mathcal{I}}_0(\mathbf{x} + \boldsymbol{\xi})\, \mathrm{d}\boldsymbol{\xi} \end{aligned} \tag{11}$$

with $\widetilde{\mathcal{I}}_0 = \mathcal{I}_0$ (reference image) or $\widetilde{\mathcal{I}}_0 = \mathcal{I}'_0 = \mathcal{I}_0 \circ \boldsymbol{\phi}$ (deformed image).

While it is possible to compute integrals as the one of Eq. (11) with numerical series when $\widetilde{\mathcal{I}}_0$ is made of a unique disk (see for instance [40]), to the best of our knowledge there is no available formula to compute integrals such as I when $\widetilde{\mathcal{I}}_0$ is made of several disks, possibly overlapping or deformed (as in the $\mathcal{I}_0'$ case). The control of the numerical accuracy of the integration scheme is crucial here. Estimating the error of classic cubature formulas requires to bound derivatives of the integrand, which is not possible here. Recent advances on probabilistic integration [15] generalize the standard Monte Carlo approach and give a probability distribution for the numerical error, but it is not clear to us whether the computational cost required by adaptive sampling is worth the gain in convergence speed. We simply propose to estimate I with the classic Monte Carlo method.

3.2 Monte Carlo Integration

This section is a brief discussion of Monte Carlo integration in the proposed framework, which basically consists in estimating integrals as sample means. Note that, although the context is different, the authors of [34] also use Monte Carlo estimation to compute integrals for inhomogeneous Boolean models. These authors aim, however, at visual quality and not at accuracy, which permits them to drastically limit the computational burden. On the contrary, in order to obtain images well suited for the assessment of DIC algorithms, we need to carefully set the sample size, as discussed below.

Let $\mathbf{X}$ be a 2D random vector of density G_σ. The integral $I(\mathbf{x})$ in Eq. (11) is the expectation of the random variable $\widetilde{\mathcal{I}}_0(\mathbf{x}+\mathbf{X})$. Let $\mathbf{X}_m, m \in \{1 \dots N_{\mathrm{MC}}\}$ be N_{MC} independent identically distributed random vectors, following the Gaussian distribution G_σ. The sample mean given by

$$\widehat{I}(\mathbf{x}) = \frac{1}{N_{\mathrm{MC}}} \sum_{m=1}^{N_{\mathrm{MC}}} \widetilde{\mathcal{I}}_0(\mathbf{x}+\mathbf{X}_m) \tag{12}$$

is thus an unbiased estimator of $I(\mathbf{x})$. According to the central limit theorem, it is common in Monte Carlo methods to consider that the sample mean $\widehat{I}(\mathbf{x})$ is a Gaussian random variable of mean $I(\mathbf{x})$ and variance s^2/N_{MC}, where s^2 is the variance of the random variable $\widetilde{\mathcal{I}}_0(\mathbf{x}+\mathbf{X})$, given by

$$s^2 = E(\widetilde{\mathcal{I}}_0(\mathbf{x}+\mathbf{X})^2) - E(\widetilde{\mathcal{I}}_0(\mathbf{x}+\mathbf{X}))^2 \tag{13}$$

Since $\widetilde{\mathcal{I}}_0$ only takes 0 or 1 values, $\widetilde{\mathcal{I}}_0(\mathbf{x}+\mathbf{X})^2 = \widetilde{\mathcal{I}}_0(\mathbf{x}+\mathbf{X})$, and since the maximum of the numerical function $f(x) = x - x^2$ is attained at $x = 1/2$, we have $s^2 \leq 1/4$.

The gray level $\mathcal{I}(\mathbf{x})$ is obtained by quantizing $\gamma \widehat{I}(\mathbf{x})$ over b bits (thus 2^b available values). Since $\gamma I(\mathbf{x})$ takes values between 0 and 1, the quantization scheme consists in assigning $\gamma \widehat{I}(\mathbf{x})$ to $k/2^b$ as soon as it is in the interval $[k/2^b, (k+1)/2^b[$, with $k \in \{0, \dots, 2^b - 1\}$. Since the sample mean $\widehat{I}(\mathbf{x})$ shows random fluctuations, $\gamma \widehat{I}(\mathbf{x})$ may be assigned to another discrete value than the one corresponding to the quantized $\gamma I(\mathbf{x})$, yielding quantization error. Using the model of "Appendix A" (that is, assuming that $\gamma I(\mathbf{x})$ is uniformly distributed over the integration interval), we are able to compute the probability $\mathcal{E}$ of such a quantization error. Here $\gamma \widehat{I}(\mathbf{x})$ is considered as a Gaussian random variable of mean $\gamma I(\mathbf{x})$ and variance $\gamma^2 s^2/N_{\mathrm{MC}}$; Eq. (36) in "Appendix A", with $b - a$ replaced by the quantization step 2^{-b}, gives the following expression of $\mathcal{E}$:

$$\mathcal{E} = \frac{\sqrt{2} 2^b \gamma s}{\sqrt{\pi N_{\mathrm{MC}}}}. \tag{14}$$

Imposing an upper bound α on the quantization error probability $\mathcal{E}$ thus gives

$$N_{\mathrm{MC}} \geq \frac{2^{2b+1}\gamma^2 s^2}{\pi \alpha^2}. \tag{15}$$

In practice, we set

$$N_{\mathrm{MC}} = \left[\frac{2^{2b+1}\gamma^2 s^2}{\pi \alpha^2} \right] \tag{16}$$

where $[x]$ is the integer part of any real number x.

Setting N_{MC} in such a way ensures that the random fluctuations caused by the stochastic nature of the estimation process are canceled out by rounding, at least for a proportion $1-\alpha$ of the pixels. This ensures that the digital image $\mathcal{I}$ really satisfies Eq. (9), except for a proportion α of the pixels which may be affected by quantization error. Parameter α shall be adjusted to control the quantization error, while keeping Monte Carlo integration tractable, N_{MC} being inversely proportional to α^2. This is discussed further in Sect. 3.3.5.

3.3 Practical Implementation

In this section, we discuss the practical implementation of the proposed speckle rendering algorithm.

3.3.1 Algorithm

Since we would like the final b-bit image to have integer values between 1 and 2^b (corresponding to the range of the gray levels), the ultimate output of the proposed algorithm is the value of $2^{b-1} + \gamma 2^b (0.5 - \widehat{I}(\mathbf{x}))$ at any pixel $\mathbf{x}$ (so that ink droplets appear as a dark pattern on a bright background), quantized over the values $\{1, \dots, 2^b\}$, γ being the contrast parameter.

Algorithm 1 renders the digital image $\mathcal{I}$ (reference image) or $\mathcal{I}'$ (deformed image under a prescribed mapping $\boldsymbol{\phi}$). Note that no interpolation scheme is ever used, which is the main motivation for which this approach is proposed and tested. Figure 2 shows four synthetic speckle images, illustrating the role of the parameters governing either the digital sensor (size $X \times Y$, bit depth b, and PSF standard deviation σ) or the speckle (contrast γ, intensity λ, and mean radius μ).

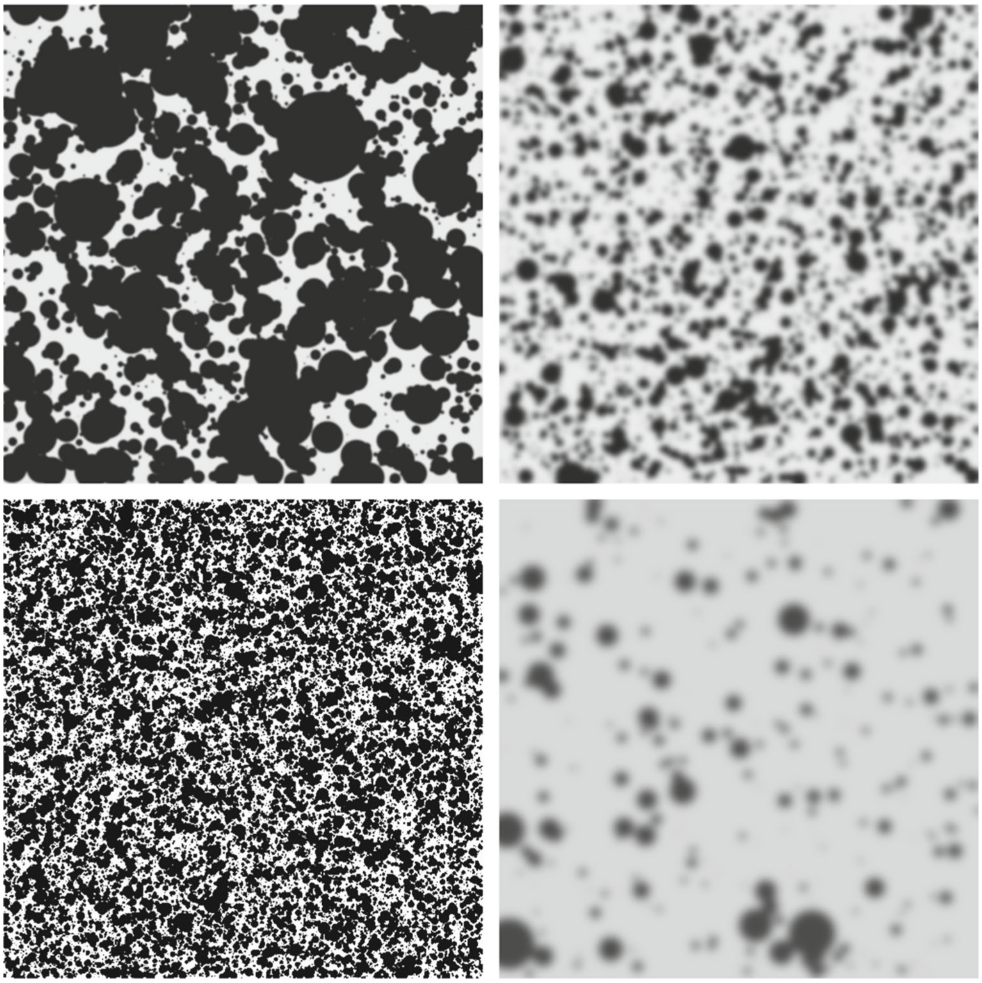


Fig. 2 8-bit 500 × 500 speckle images. From left to right and from top to bottom: Experiments 1 to 4. Parameters are given in Table 2

Table 2 Sensor and speckle parameters for Experiments 1 to 4

Exp.	γ	σ (pixel)	λ (no. of disks per image)	μ (pixel)	Covering κ (Eq. (6)) (%)	Perimeter θ (Eq. (7))
1	0.8	1	2000	5	72	3576
2	0.8	3	4000	2	33	16,811
3	0.95	0.1	40,000	1	63	91,968
4	0.6	5	300	8	38	1163

Some of these patterns may not look realistic in the context of experimental mechanics, but they are chosen here for the sake of pedagogy. The parameters of these four numerical experiments are given in Table 2. The numerical value of the perimeter θ has to be compared to the perimeter of the image domain Ω, equal here to 2000 pixels. No deformation is used here (that is, $\boldsymbol{\phi} = \mathrm{Id}$ is the identity mapping). Three examples of 300×300 deformed speckle images can be seen in Fig. 3. The amplitude of the deformations is voluntarily large for a better visibility; this is not representative of typical deformations encountered in experimental mechanics which are much smaller.

3.3.2 Computational Speedups

Several speedups are used in the actual implementation of Algorithm 1.

Variance estimation In line 4, the variance s^2 is estimated at every pixel $\mathbf{x}$ from a sample of size N_0 (here, $N_0 = 500$), by $s^2(\mathbf{x}) = \widehat{I}(\mathbf{x}) - \widehat{I}^2(\mathbf{x})$ where $\widehat{I}(\mathbf{x})$ is the sample mean. This online estimation adapts the sample size N_{MC} according to the value of $I(\mathbf{x})$: N_{MC} is maximum if $I(\mathbf{x})$ is close to $1/2$ (corresponding to pixels lying near the boundary of the disks $\mathbf{D}_i$), and it is minimum if $I(\mathbf{x})$ is close to 0 or 1. Note

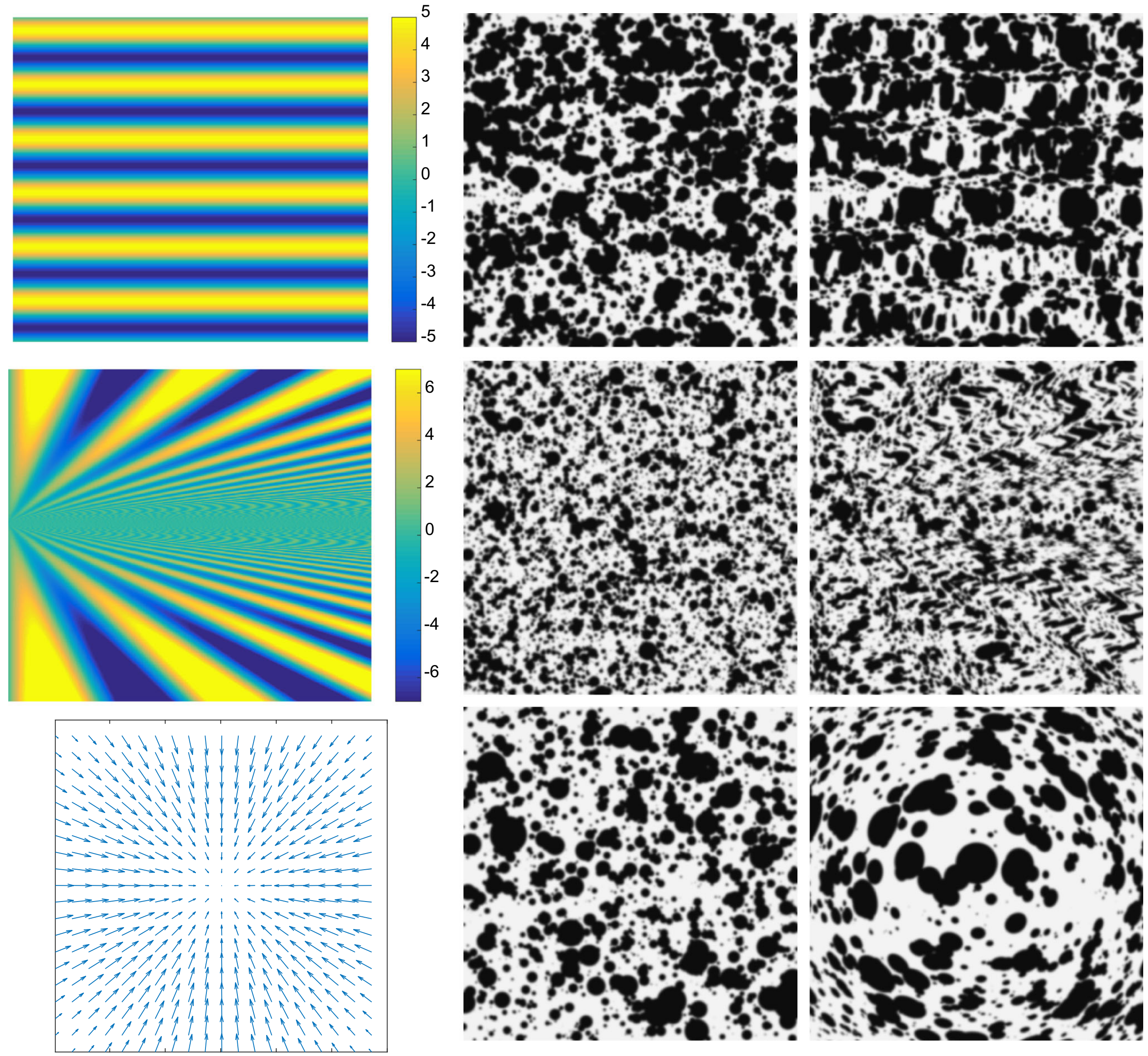


Fig. 3 From the left to the right: inverse displacement field $\mathbf{U}$, reference (non-deformed), and deformed speckle images. First row: $\mathbf{U}_x$ (displayed) is a sine wave of amplitude 5 and period 50 pixels, and $\mathbf{U}_y = 0$. Second row: $\mathbf{U}_y$ (displayed) is such that the derivative along y (corresponding to the strain component ε_{yy}) is a sine wave of amplitude .5 along y and of period decreasing with x, and $\mathbf{U}_x = 0$. Third row: $\mathbf{U}$ flow is defined as the "punch" function of [37]. Here, x-axis and y-axis denote vertical and horizontal axis, respectively

that large PSF (i.e., large σ) with respect to the size of the disks are more likely to give pixels whose gray value is not close to 0 or 1, thus larger computation time. Such an online estimation is all the more interesting as N_{MC} is proportional to 2^{2b} and can attain very large values. Indeed, conservatively setting N_{MC} to its maximum value would necessitate, for every pixel, from 330,000 iterations for an 8-bit image to 85 millions iterations for a 12-bit image, and to 22 billions iterations for a 16-bit image (with $\gamma = 1$, $s^2 = 1/4$, and $\alpha = 0.1$ in Eq. (16)).

Testing whether $\boldsymbol{\phi}(\mathbf{x} + \mathbf{X}_m)$ belongs to P. In line 7, testing whether a point $\boldsymbol{\phi}(\mathbf{x}+\mathbf{X}_m)$ belongs to one of the disks $\mathbf{D}_i$ may be time-consuming, because the number N of disks can be as high as several thousands and this test must be performed for any m in $\{1, \ldots, N_{\text{MC}}\}$ and for any pixel $\mathbf{x}$. We propose the following speedup, based on the fact that, in practice, strain fields, thus displacement gradients, have a limited amplitude. For any pixel $\mathbf{x}$, we pre-compute the list $\mathcal{L}(\mathbf{x})$ of the indices of disks to which $\boldsymbol{\phi}(\mathbf{x}+\mathbf{X}_m)$ may belong. This list is defined as the list of disk indices whose center is below a given distance

```
Data: Speckle parameters (expected number of disks in the
      resulting image $\lambda$, expected radius of a disk $\mu$, contrast $\gamma$),
      camera parameters (standard deviation $\sigma$ of the PSF, size
      of the resulting $X \times Y$ image, quantization depth $b$),
      quantization error probability $\alpha$, mapping $\boldsymbol{\phi}$ ($\boldsymbol{\phi} = Id$ when
      rendering a reference image)
Draw the number of disks $N$, following a Poisson distribution of
intensity $\lambda$;
Draw $N$ disk centers $\mathbf{d}_i$ uniformly distributed in the image
domain, and $N$ radii $R_i$ following an exponential distribution of
mean $\mu$;
for any pixel $\mathbf{x} \in \{1 \dots X\} \times \{1 \dots Y\}$ do
    Estimate $s^2$ and set $N_{\text{MC}} = 2^{2b+1}\gamma^2 s^2/(\pi^2\alpha^2)$;
    for $m \in \{1, \dots, N_{MC}\}$ do
        Draw $\mathbf{X}_m$ following a 2D Gaussian distribution of
        standard deviation $\sigma$;
        Define $\mathcal{I}_0(\boldsymbol{\phi}(\mathbf{x}+\mathbf{X}_m)) = 1$ if $\boldsymbol{\phi}(\mathbf{x}+\mathbf{X}_m)$ belongs to a
        disk centered at one of the $\mathbf{d}_i$ of radius $R_i$, and
        $\mathcal{I}_0(\boldsymbol{\phi}(\mathbf{x}+\mathbf{X}_m)) = 0$ otherwise.
    end
    Calculate $\widehat{I}(\mathbf{x}) = \frac{1}{N_{\text{MC}}}\sum_{m=1}^{N_{\text{MC}}} \mathcal{I}_0(\boldsymbol{\phi}(\mathbf{x}+\mathbf{X}_m))$;
end
Quantization: the resulting digital image is
$\mathcal{I}(\mathbf{x}) = [2^{b-1} + \gamma 2^b(0.5 - \widehat{I}(\mathbf{x}))]$ (the resulting gray level range
is $\{1, \dots, 2^b\}$);
Result: a digital image $\mathcal{I}$.
```

Algorithm 1: Speckle image rendering (sketch of the algorithm). In practice, speedups described in Sect. 3.3.2 are used.

to $\boldsymbol{\phi}(\mathbf{x})$, namely:

$$\mathcal{L}(\mathbf{x}) = \{i \in \{1, \dots n\}, ||\boldsymbol{\phi}(\mathbf{x}) - \mathbf{d}_i|| \leq R_i + (1+\delta)B\} \tag{17}$$

where $||\cdot||$ is the Euclidean norm, δ is a uniform upper bound of the Frobenius norm of the Jacobian matrix $\mathbf{J}$ of the inverse displacement field $\mathbf{U}$ (such that the mean value inequality writes $||\mathbf{U}(\mathbf{x}) - \mathbf{U}(\mathbf{y})|| \leq \delta\, ||\mathbf{x} - \mathbf{y}||$ for any $(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^2 \times \mathbb{R}^2$), and the random vectors $\mathbf{X}_m$ are assumed for a while to satisfy $||\mathbf{X}_m|| \leq B$ for any $m \in \{1, \dots, N_{\text{MC}}\}$, with $B > 0$. Note that $\delta = 0$ when rendering the reference image for which $\mathbf{U} = \mathbf{0}$.

Indeed, since

$$\begin{aligned}\boldsymbol{\phi}(\mathbf{x}) - \mathbf{d}_i &= (\boldsymbol{\phi}(\mathbf{x}) - \mathbf{x}) + (\mathbf{x} + \mathbf{X}_m - \boldsymbol{\phi}(\mathbf{x}+\mathbf{X}_m)) \\ &\quad + (\boldsymbol{\phi}(\mathbf{x}+\mathbf{X}_m) - \mathbf{d}_i) - \mathbf{X}_m,\end{aligned} \tag{18}$$

$\boldsymbol{\phi}(\mathbf{x}) - \mathbf{x} = \mathbf{U}(\mathbf{x})$, and $\mathbf{x} + \mathbf{X}_m - \boldsymbol{\phi}(\mathbf{x}+\mathbf{X}_m) = -\mathbf{U}(\mathbf{x}+\mathbf{X}_m)$ (by definition of mapping $\boldsymbol{\phi}$ and inverse displacement $\mathbf{U}$), we have:

$$\begin{aligned}\boldsymbol{\phi}(\mathbf{x}) - \mathbf{d}_i &= (\mathbf{U}(\mathbf{x}) - \mathbf{U}(\mathbf{x}+\mathbf{X}_m)) + (\boldsymbol{\phi}(\mathbf{x}+\mathbf{X}_m) - \mathbf{d}_i) \\ &\quad -\mathbf{X}_m.\end{aligned} \tag{19}$$

By definition of B, $||\mathbf{X}_m|| \leq B$, and by the mean value inequality, the following inequality holds:

$$||\mathbf{U}(\mathbf{x}) - \mathbf{U}(\mathbf{x}+\mathbf{X}_m)|| \leq \delta B. \tag{20}$$

Therefore, it follows from Eq. (19) and the triangle inequality that:

$$||\boldsymbol{\phi}(\mathbf{x}) - \mathbf{d}_i|| \leq (1+\delta)B + ||\boldsymbol{\phi}(\mathbf{x}+\mathbf{X}_m) - \mathbf{d}_i||. \tag{21}$$

Consequently, if $\boldsymbol{\phi}(\mathbf{x}+\mathbf{X}_m)$ belongs to the disk $\mathbf{D}_{i_0}$, then $||\boldsymbol{\phi}(\mathbf{x}+\mathbf{X}_m) - \mathbf{d}_{i_0}|| \leq R_{i_0}$ and $i_0 \in \mathcal{L}(\mathbf{x})$. This means that the indices of the disks in which $\boldsymbol{\phi}(\mathbf{x}+\mathbf{X}_m)$ may fall belong to $\mathcal{L}(\mathbf{x})$ for every m. This ensures that no disk is missed when restricting to $\mathcal{L}(\mathbf{x})$ the search for disks to which $\boldsymbol{\phi}(\mathbf{x}+\mathbf{X}_m)$ may belong.

Now, since $\mathbf{X}_m$ is a Gaussian vector, $||\mathbf{X}_m||$ is not uniformly bounded. This quantity has, however, a probability smaller than p to be larger than $B = G^{-1}(1-p)\sigma$, where G is the cumulative distribution function of the standard normal law. Even if highly unlikely, $\boldsymbol{\phi}(\mathbf{x}+\mathbf{X}_m)$ such that $||\mathbf{X}_m|| > B$ may erroneously give $\widetilde{\mathcal{I}}_0(\boldsymbol{\phi}(\mathbf{x}+\mathbf{X}_m)) = 0$ instead of 1. On average, $p\,N_{\text{MC}}$ samples may be erroneously evaluated in the sample mean, then giving an average error of p on the estimation of $\widehat{I}$. We thus choose p equal to a tenth of the standard deviation $s/(\sqrt{N_{\text{MC}}}) = \pi\alpha/(\sqrt{2}2^b)$ of $\widehat{I}(\mathbf{x})$, so that the bias p is well below the random fluctuations of the sample mean $\widehat{I}(\mathbf{x})$. With $\alpha = 0.1$, this yields for $b = 8$, $B = G^{-1}(1-p)\sigma = 3.75\,\sigma$, for $b = 12$, $B = G^{-1}(1-p)\sigma = 4.40\,\sigma$, and for $b = 16$, $B = G^{-1}(1-p)\sigma = 4.97\,\sigma$.

Since $\mathcal{L}(\mathbf{x})$ is significantly smaller than the number N of disks (its cardinality is typically below 5 and is independent of the image size), the calculation time of the Monte Carlo estimation is significantly reduced by pre-computing $\mathcal{L}(x)$ for any pixel $\mathbf{x}$. This approach is relevant for limited strain (δ is usually below some percents), since the cardinality of $\mathcal{L}(x)$ increases with δ.

A remark on unbounded strain fields Without any additional assumption, if the strain field is not bounded, $\Phi(\mathbf{x}+\mathbf{X}_m)$ may belong to any disk $\mathbf{D}_i$ and building $\mathcal{L}(\mathbf{x})$ would be useless. However, when considering unbounded strain yet bounded displacement, we have:

$$||\mathbf{U}(\mathbf{x}) - \mathbf{U}(\mathbf{x}+\mathbf{X}_m)|| \leq 2M \tag{22}$$

where M is an upper bound to the displacement field. In this case, the list $\mathcal{L}(\mathbf{x})$ satisfies:

$$\mathcal{L}(\mathbf{x}) = \{i \in \{1, \dots n\}, ||\boldsymbol{\phi}(\mathbf{x}) - \mathbf{d}_i|| \leq R_i + 2M + B\} \tag{23}$$

instead of the definition given by Eq. (17).

Parallel execution Sample mean calculation is parallelized by running lines 5 to 8 on a Graphics Processing Unit (GPU).

Pseudorandom sample The authors of [34] use the same random sequence $(\mathbf{X}_m)_{1\leq m\leq N_{\mathrm{MC}}}$ for any pixel $\mathbf{x}$, which slightly improves the computation time. We do not adopt this simplification in order to avoid the results to be affected by a systematic bias. Generally speaking, we do not use any heuristic speedup which could bias subsequent statistics estimated from the speckle images.

3.3.3 Influence of the Parameters on the Calculation Time

Apart from memory management, the calculation time of a serial execution of Algorithm 1 is proportional to the number of pixels in the image, and to the size of the Monte Carlo sample. This sample size is proportional to $1/\alpha^2$ and contrast γ, and it increases with the bit depth b, the size being proportional to 2^{2b}. As mentioned above, it also increases with σ, large-support PSF giving a sample variance s^2 close to the maximum value $1/4$ for a larger number of pixels. Nevertheless, typical values for σ are below 1 pixel, as lenses used for mechanical tests are sharp. The evolution of the calculation time of a parallel execution does not strictly follow these guidelines as it also depends on hardware considerations.

Monte Carlo estimation is well suited to an execution on a parallel architecture (either on a multi-processor / multi-core system or on a GPU): our software implementation of the proposed algorithm is a MATLAB 2016A code with parallelization on a GPU, using single-precision floating point numbers (cf. the discussion of Sect. 3.3.4). Table 3 gives the computation times for rendering the 500 × 500 images of the experiments of the previous section. Note that rendering 1000 × 1000 images would require four times these calculation times. Two desktop computers have been used for this test. Configuration 1 is equipped with a 4-core Intel Xeon E3-1270 v3 @3.50GHz CPU with 32 Gb memory and a Nvidia Quadro K2000 graphic card with 2 Gb memory. Configuration 2 is equipped with a 6-core Intel Xeon E5-1650 v4 @ 3.60GHz CPU with 64 Gb memory and a NVidia Titan X graphic card with 12 Gb memory. The CPU implementation is parallelized through multi-core processing. The 8-bit non-deformed speckle images of Fig. 3 require between 2.8 and 4.3 min on Desktop 2. Note that our code is likely to run faster in the near future, thanks to rapid improvements of MATLAB and of the computing power of GPU. Our software implementation would benefit greatly from a native CUDA code or from exploiting large-scale parallel computing resources, especially for 16-bit images. Nevertheless, it should be noted that calculation time is not really critical in the considered application: the ultimate goal of this work is to produce once and for all several ground-truth datasets for DIC assessment.

Table 3 Calculation times in minutes for rendering 500 × 500 images

Experiment no.	Time (min)		
	Config. 1		Config. 2
	CPU	GPU	GPU
1 (8-bit, $\alpha = 0.05$)	40.3	14.5	5.0
2 (8-bit, $\alpha = 0.05$)	146.5	71.2	24.3
3 (8-bit, $\alpha = 0.05$)	28.6	11.3	3.6
4 (8-bit, $\alpha = 0.05$)	32.6	14.2	8.0
1 (8-bit, $\alpha = 0.1$)	10.7	4.9	3.6
2 (8-bit, $\alpha = 0.1$)	38.7	24.2	17.5
3 (8-bit, $\alpha = 0.1$)	8.1	4.0	2.9
4 (8-bit, $\alpha = 0.1$)	8.5	6.1	5.8
1 (12-bit, $\alpha = 0.1$)	3132*	1063*	73.6
2 (12-bit, $\alpha = 0.1$)	9427*	4820*	360.4
3 (12-bit, $\alpha = 0.1$)	2101*	589*	57.0
4 (12-bit, $\alpha = 0.1$)	1990*	877*	69.5
1 (16-bit, $\alpha = 0.1$)	–	–	14,214**
2 (16-bit, $\alpha = 0.1$)	–	–	66,875**
3 (16-bit, $\alpha = 0.1$)	–	–	11,789**
4 (16-bit, $\alpha = 0.1$)	–	–	17,390**

Asterisks mark extrapolated times based on the time needed for rendering 5000 pixels, double asterisks based on 20 pixels

All other things being equal, increasing σ increases the calculation time, as discussed above. It depends, however, on the perimeter of the speckle pattern: relatively large σ and large perimeter values (as in Experiment 2) give larger computation times than relatively large σ and small perimeter values (as in Experiment 4), since, in this latter case, much less pixels $\mathbf{x}$ lie at the border of a disk and have an average gray value.

3.3.4 Single-Precision or Double-Precision Calculation?

It should be noted that an accurate calculation of the sample mean requires some attention. Although it is difficult to assess thoroughly, the effect of rounding errors due to the machine representation of real values [23], the aim of this section is to discuss whether single-precision floating point representation is sufficient for the present problem. This question is important in practice, since memory occupation is divided by two when using single-precision instead of double-precision, and, chiefly, most GPU process single-precision floats much faster.

In Eq. (12), computing the sample mean basically consists in dividing two integer numbers. If the division and the number representation follow IEEE-754 standard (implemented in MATLAB), the sample mean is thus determined up to the "machine epsilon", which is $\varepsilon_s = 2^{-23}$ in single precision and $\varepsilon_s = 2^{-52}$ in double precision [17,23]. The machine epsilon is well below the quantization step $1/2^b$ even when 16-bit images are rendered.

However, the sample size $N_{\mathrm{MC}} = [\gamma^2 s^2 2^{2b+1}/\pi^2\alpha^2]$, can be as large as $10 \cdot 2^{2b-1}$ when $\alpha = 0.1$. The sample size and the summation involved in the sample mean must be performed with a datatype adapted to this range. Integers below $2^{24} = 16{,}777{,}216$ being represented exactly by a single-precision value, the summation involved in the sample mean can be calculated safely when $b = 8$, since $10 \cdot 2^{2b-1} = 327{,}680$ in this case. Sample sizes above 2^{24} likely give, however, rounding errors. Such a value is attained for $b = 12$. The problem can be avoided by rearranging the summation, which must be done in practice.

The parallel calculation of the sample mean requires indeed that the whole series $(\mathbf{X}_m)$ of random vectors is generated on the GPU and fits its memory. Such a single-precision $2 \times N_{\mathrm{MC}}$ vector would require around 5 Mb of memory for N_{MC} corresponding to $b = 8$, around 1.2 Gb for $b = 12$, and 1 million Gb for $b = 16$. It is thus obviously needed to split the sample and to calculate the sample mean by summation of the results obtained on the subsamples. We actually perform the calculation of the sample mean by computing the average value of K means m_k over subsamples of size N such that $N_{\mathrm{MC}} = KN$. We compute independently the K terms of the following sum:

$$\widehat{I} = \sum_{k=1}^{K} \frac{N}{N_{\mathrm{MC}}} m_k \tag{24}$$

with

$$m_k = \frac{1}{N} \sum_{m=1}^{N} \widetilde{\mathcal{I}}_0(\mathbf{x} + \mathbf{X}_m). \tag{25}$$

Setting $N \leq 2^{24}$ ensures that the summation in Eq. (25) is not affected by rounding errors. Note that $N \leq 2^{24}$ may be required in order that $(X_m)_{1 \leq m \leq N}$ and auxiliary variables fit the GPU memory. As discussed above, each m_k is determined up to ε_s, thus, in the worst case, $N/N_{\mathrm{MC}} m_k$ too. $\widehat{I}$ being the sum of K term determined up to $2\varepsilon_s$, it is determined up to $2K\varepsilon_s$. For instance, $N = 2^{23}$ is required to fit a 2 Gb GPU memory. In this case, $K = N_{\mathrm{MC}}/N \simeq 10 \cdot 2^{2b-22}$ is at most around $10 \cdot 2^2$ for a 12-bit image, thus the numerical error $2K\varepsilon_s$ of $\widehat{I}$ is at most around $10 \cdot 2^{-21}$ and is well below the quantization step $1/2^{-12}$. On the contrary, a 16-bit quantization would give $2K\varepsilon_s$ at most around $10 \cdot 2^{-10}$ which is above the quantization step $1/2^{16}$. In this case, rounding error in single-precision calculation may cause quantization error. It should, however, be noted that this calculation corresponds to a worst case, in which all of the m_k in the sum of (24) are affected by a rounding error. Compensated summation may also be used in practice [23].

We can conclude that all calculations can be led safely with single-precision floating point numbers when rendering 8-bit or 12-bit images. Concerning 16-bit images, the worst-case analysis shows that it would be safer to lead the calculations with double-precision floats. Whether this is really a problem in real-world rendering requires, however, further investigation.

To conclude this discussion on practical aspects of the software implementation, let us mention that the period of the pseudorandom generator must be long enough in order that the random sample $(\mathbf{X}_m)_{1 \leq m \leq N_{\mathrm{MC}}}$ generated at any pixel $\mathbf{x}$ can still be considered as a realization of an independently distributed process. The period of the generators implemented in MATLAB seems to be well above what is needed.

3.3.5 Rendering Accuracy Assessment

In Sect. 3.2, setting the sample size by bounding the error probability by α means that the numerical accuracy of the estimation of I is expected to be below the quantization step for a proportion of the pixels at most equal to $1 - \alpha$.

Nevertheless, some assumptions permitting to establish Eq. (15) may not be valid. For example, the calculation of "Appendix A" assumes that the sample mean is normally distributed. Central limit theorem backs this hypothesis; this is, however, only an asymptotic result. Bounding $||\mathbf{X}_m||$ by B in Sect. 3.3.2 may also introduce a bias. Moreover, the implementation of Algorithm 1 relies on a pseudorandom generator which only approximates the notion of realization of independent variables.

We, therefore, perform a sanity check of the proposed algorithm. Of course, it is not possible to measure the numerical accuracy (i.e., the discrepancy between the estimated gray level $\widehat{I}$ which may show random fluctuations, and the deterministic theoretical gray level $\mathcal{I}$ which is unknown) based on the rendering of a single image. The difference $\delta(\mathbf{x})$ between two images rendered by two independent Monte Carlo estimations (after quantization on b bits), run on the same infinite resolution speckle image, should, however, be different from 0 for a proportion of pixels smaller than 2α.

We have calculated this difference for the images of Fig. 2 and various bit depths. Table 4 gathers the obtained results. We can see that the proportion of pixels in the difference image δ affected by quantization errors (i.e., such that $\delta(\mathbf{x}) \neq 0$) is below 2α, as expected from the theory. The result still holds for the strong (inverse) displacement field corresponding to Experiments 1b to 4b. This confirms that at most a proportion α of pixels from the rendered digital image $\mathcal{I}$ may be affected by quantization error. Moreover, this experiment also shows that the vast majority of these erroneous pixels have a gray value error of 1. If the user generates noisy images (as in Sect. 3.5), quantization errors are

Table 4 From left to right: experiment number, bit depth of the render image, probability error α, proportion of pixels $\mathbf{x}$ such that the difference $\delta(\mathbf{x})$ between two quantized independent Monte Carlo estimation is different from 0, number of pixels such that $|\delta(\mathbf{x})| = 1$, and number of pixels such that $|\delta(\mathbf{x})| = 2$

Experiment			% pixels	# pixels	# pixels
No.	bit	α	s.t. $\delta \neq 0$	s.t. $\lvert\delta\rvert = 1$	s.t. $\lvert\delta\rvert = 2$
1	4	0.1	2.4	6034	0
2	4	0.1	9.2	22,894	0
3	4	0.1	1.4	3552	0
4	4	0.1	2.3	5774	0
1	8	0.05	2.6	6525	0
2	8	0.05	7.0	17,521	0
3	8	0.05	1.5	3679	0
4	8	0.05	4.9	12,126	0
1	8	0.1	5.1	12,666	0
2	8	0.1	13.9	34,692	1
3	8	0.1	2.9	7294	0
4	8	0.1	9.6	24,052	0
1	12	0.1	4.95	12,369	0
2	12	0.1	14.2	35,550	0
3	12	0.1	2.9	7305	1
4	12	0.1	9.8	24,414	2
1b	4	0.1	2.6	6676	0
2b	4	0.1	9.2	23,062	0
3b	4	0.1	1.4	3600	0
4b	4	0.1	2.6	6530	0
1b	8	0.05	2.0	5105	0
2b	8	0.05	6.4	16,017	0
3b	8	0.05	1.2	2971	0
4b	8	0.05	3.9	9705	0
1b	8	0.1	3.3	8323	0
2b	8	0.1	12.8	32,112	0
3b	8	0.1	2.4	6119	0
4b	8	0.1	7.3	18,160	0
1b	12	0.1	3.8	9566	2
2b	12	0.1	14.1	35,407	0
3b	12	0.1	2.4	6025	0
4b	12	0.1	7.7	19,257	0

In all these experiments, $|\delta(\mathbf{x})|$ is never larger than 2. The proportion of pixels has to be compared to 2α, and the numbers of pixels have to be compared to the 500 × 500 pixels in the images. Several bit depths b have been tested. Experiments 1b to 4b have the same parameters as Experiments 1 to 4, but the "punch" function of Fig. 3 is used

likely to be much below the noise floor. Finally, Fig. 4 shows the difference image δ for $\alpha = 0.05$. We can see that the pixels affected by quantization error mainly lie at the borders of the black disks.

Since this experiment suggests that setting $\alpha = 0.1$ actually gives at most 5% of pixels affected by quantization error (except when σ has a quite large value, as in Experiment 2 where the error probability is still bounded by α), we recommend to use this value when computing the sample size N_{MC}.

3.3.6 Comparing the Proposed Rendering Method with Interpolation and Pixel Binning

As mentioned in the introduction, some authors render the speckle image of a deformed state by interpolating a high-resolution, either real or synthetic, speckle image under the prescribed displacement, and make use of pixel binning to reduce interpolation error. How much pixels should be binned together with respect to the interpolation scheme is, to the best of our knowledge, an overlooked problem which has not yet a rigorous solution. A full comparison of both methods in the context of the assessment of DIC algorithms is beyond the scope of this paper. Comparing the retrieved displacement field with respect to the prescribed displacement field indeed depends on various factors such as the systematic bias modeled by a Savitzky–Golay filter [43], error caused by the interpolation scheme needed by DIC itself [7,8], the shape of the speckle patterns [29,30], and quantization noise. It is thus difficult to isolate the error solely caused by the process rendering the input speckle images. We propose, however, a numerical experiment suggesting that the proposed method is much more reliable than interpolation and pixel binning.

We render two 1024 × 1024 8-bit speckle images with the proposed algorithm, one, $\mathcal{I}$, for the reference state and the other, $\mathcal{I}'$, for the deformed state. The value of α is such that at most 5% of the pixels have an incorrect gray value, as discussed in the preceding section. The prescribed displacement field $\mathbf{U}$ is such that $\mathbf{U}_y(x, y)$ is a sine function of x of amplitude 0.5 pixel and of period linearly increasing from 10 to 150 pixels along the y-axis, and $\mathbf{U}_x = 0$. We also render an image of the deformed state $\overline{\mathcal{I}'}$ with interpolation, either by a linear or cubic scheme. We define the binning operator $\mathcal{B}_p$ such that $\mathcal{B}_p(\mathcal{I})$ is obtained from any image $\mathcal{I}$ by substituting each $p \times p$ array of pixels in $\mathcal{I}$ by its mean value, giving the image $\mathcal{B}_p(\mathcal{I})$ whose dimensions are p times smaller than the input image $\mathcal{I}$. Pixel binning reduces the gray value error, resulting from the α parameter in the proposed method or from the interpolation scheme, at the price of a reduced spatial resolution. For example, $\mathcal{B}_8(\mathcal{I}')$ is a 128 × 128 image. Since the proposed rendering method has a controlled error, still reduced by pixel binning, its output can be considered as ground-truth data. In order that the binned images $\mathcal{B}_p(\mathcal{I})$ and $\mathcal{B}_p(\mathcal{I}')$ show the same intensity distribution and carry the same amount of information, we render the 1024 × 1024 images $\mathcal{I}$ and $\mathcal{I}'$ such that the random disks have an average radius of p (thus $\mathcal{B}_p(\mathcal{I})$ and $\mathcal{B}_p(\mathcal{I}')$ show disks of average radius 1 for any value of p). The number of disks is such that the covering is 50% (cf. Table 1), and the Gaussian PSF has a standard deviation of 1 pixel.

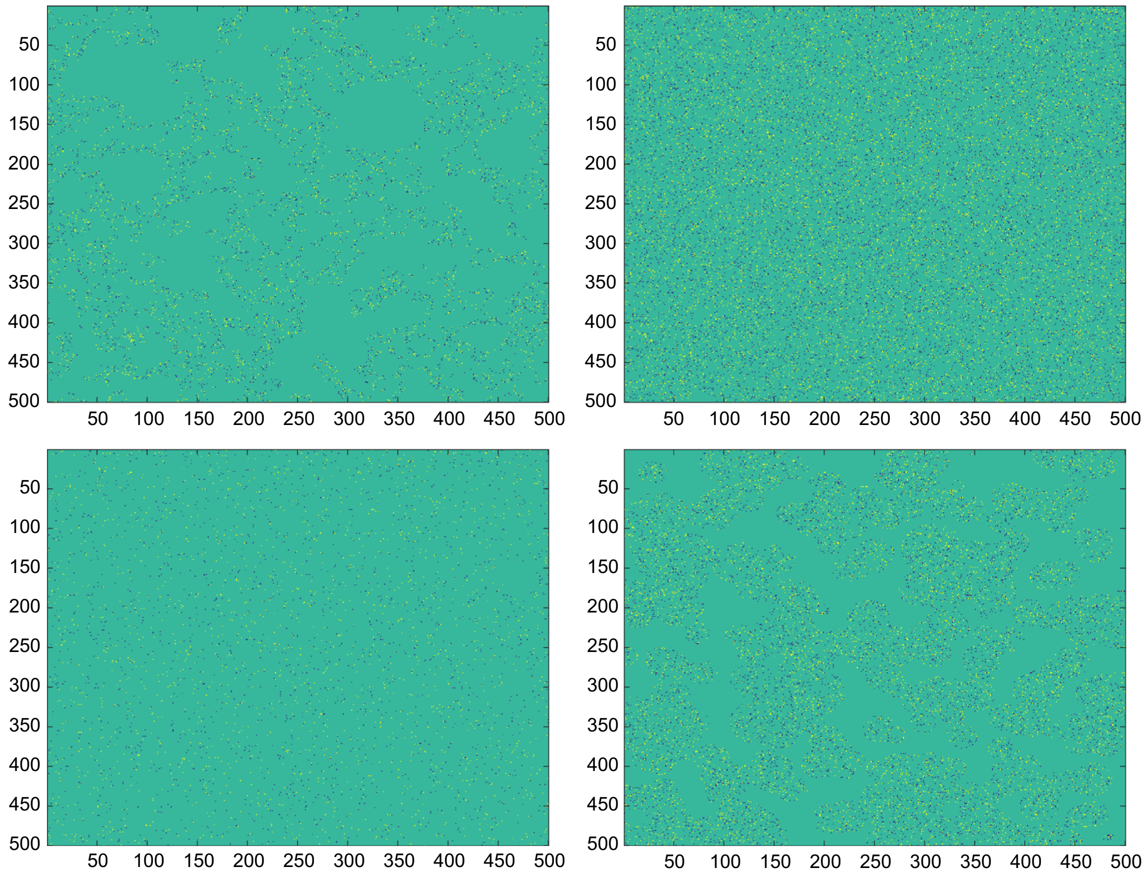


Fig. 4 Difference image δ for Experiments 1 to 4 of Fig. 2 (8-bit images, $\alpha = 0.05$), from top left to bottom right. Green pixels have no quantization error, yellow pixels are such that $\delta \geq 1$, and blue pixels are such that $\delta \leq -1$. The proportion of pixels such that the difference image is different from 0 is equal to 2.6, 7.0, 1.5, and 4.9%, respectively

Figure 5 shows the frequency distribution of the absolute difference $|\mathcal{B}_p(\mathcal{I}') - \mathcal{B}_p(\overline{\mathcal{I}'})|$, for linear and cubic interpolation and p belonging to $\{1, 2, 4, 8\}$. As we can see, when no binning is used ($p = 1$), less than 40% of the pixels have the same gray value with the two methods. A significant proportion has a gray value error larger than 4. As expected, pixel binning reduces the gray value error. However, more than 20% of the pixels show a gray value error larger than 1 with a 4×4 binning scheme, either with linear or cubic interpolation, while our method was proved to have much less than 5% of spurious gray values. It can also be noticed that, unsurprisingly, cubic interpolation gives less erroneous gray values than linear interpolation. With 8×8 binning and cubic interpolation, 5% of the total amount of pixels differ between the two methods.

This numerical assessment, which needs to be completed by further studies, suggests that both methods may be hardly distinguishable when a 8×8 pixel binning scheme and cubic interpolation are used. Since this experiment relies on a specific prescribed displacement, it should not lead to a hasty generalization. Of course, interpolation coupled with binning requires very high-resolution images of the initial state in order to obtain large enough low-resolution images after binning. To our knowledge, no theoretical guaranty is available for the error of such a rendering method. On the contrary, the proposed algorithm does not require any interpolation and comes with a handy parameter, namely α, which controls the error.

3.4 Potential Extensions of the Proposed Model

Although the proposed framework permits the user to render realistic speckle images, it can still be adapted to various PSF or speckle models.

For example, it should be noted that the same calculation holds for any PSF. Since any PSF integrates to 1, it can be seen as the density of a random variable. In practice, we just need to generate the random vector X following this density to mimic the effect of any PSF function.

Concerning the Boolean model, we could use other shapes than disks. Any shape $\mathcal{S}$ can be used, provided that the indicator function $\mathbb{1}_{\mathcal{S}}(\mathbf{x})$ can be calculated. Using a bounding disk

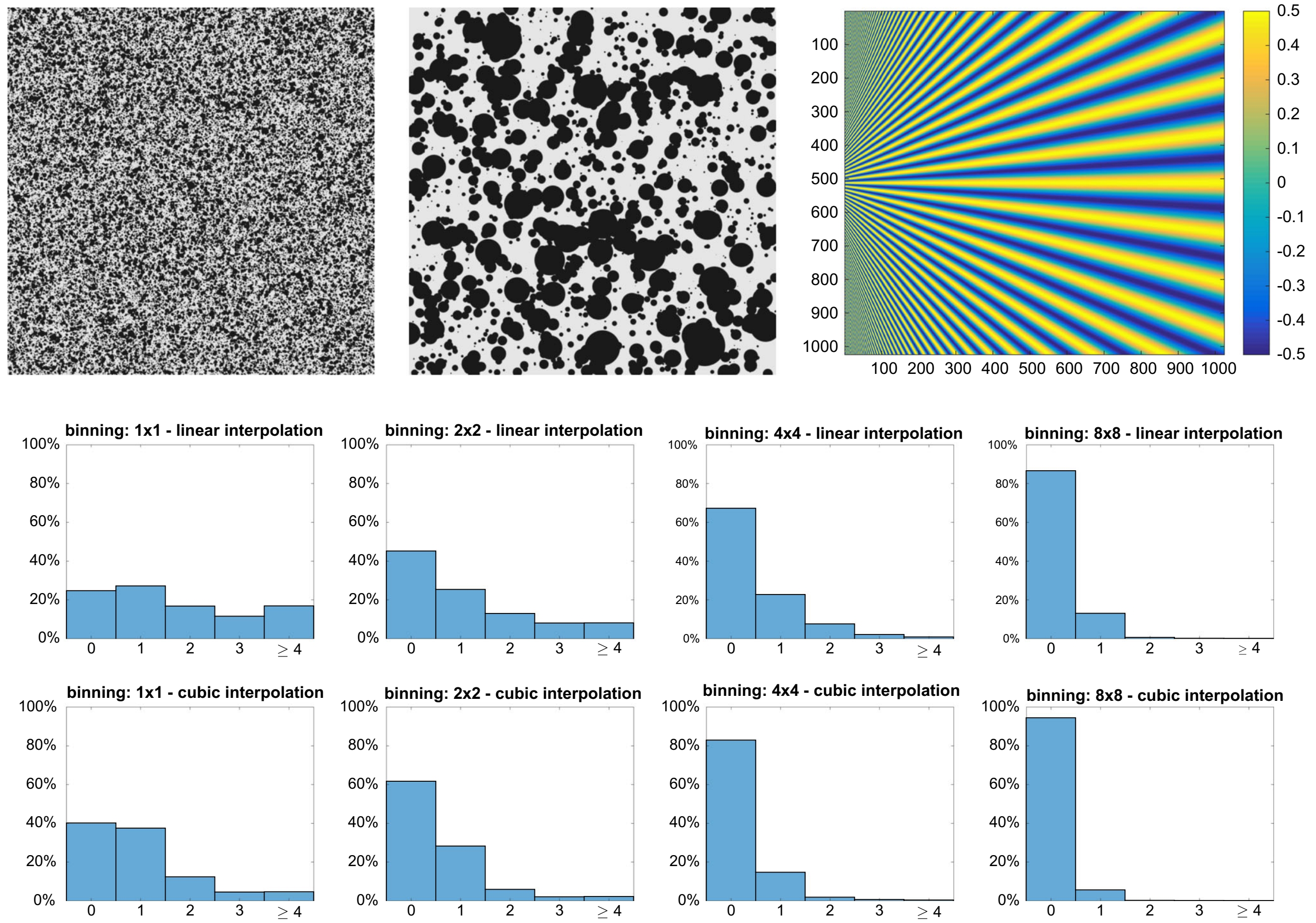


Fig. 5 Top row: the 1024 × 1024 speckle image when no binning is employed, the 1024 × 1024 speckle image before the 8 × 8 binning (both in reference state), and $\mathbf{U}_y$ displacement field over a 1024 × 1024 field. Middle and bottom rows: frequency distribution of the gray level differences (absolute value) between the deformed speckle image obtained by the proposed algorithm and by interpolation, after $p \times p$ pixel binning (middle row: linear interpolation; bottom row: cubic interpolation). For instance, approximately 55% of the pixels have the same gray level for both methods with linear interpolation and no pixel binning ($p = 1$), and approximately 20% of the pixels differ from one gray level with cubic interpolation and $p = 16$

of radius R_i would allow defining the list $\mathcal{L}(\mathbf{x})$ discussed in Sect. 3.3.2.

The Boolean model implicitly assumes that the ink projected on the specimen is not transparent: the output of the model is the same, whatever the number of overlapping disks at a given pixel. It is interesting to note that transparent ink could be modeled with the so-called transparent dead leaves model described in [21]. We do not elaborate further on this subject and leave it for future work.

3.5 Signal-Dependent Noise

Because of the quantum nature of light, any digital image sensor is affected by noise. The classical signal-dependent noise model [1,25] expresses the gray level intensity i at a given pixel $\mathbf{x}$ in a linear camera (that is, disregarding nonlinear processing such as γ-correction) as:

$$i(\mathbf{x}) = g\rho_{p(\mathbf{x})} + \tau(\eta, \nu^2) \tag{26}$$

where g is the gain, $\rho_{p(\mathbf{x})}$ is a Poisson random variable of intensity $p(\mathbf{x})$ modeling the shot noise, and τ is a Gaussian random variable of mean η (offset, set by the manufacturer) and variance ν^2. Intensity $p(\mathbf{x})$ is the number of electrons generated at $\mathbf{x}$, caused either by conversion of photons or spontaneously generated as a dark signal, and τ models various sources of noise such as readout noise. A simple calculation gives the following affine relation between the variance and the expected value of i:

$$\mathrm{Var}(i(\mathbf{x})) = gE(i) + \nu^2 - g\eta. \tag{27}$$

The importance of considering such a noise model when assessing the metrological performance of DIC in real-world experiments is discussed in [8]. Except for low-light imaging, it is common to consider an additive Gaussian white noise of variance given by Eq. (27) instead of the Poisson–Gaussian noise of Eq. (26).

To mimic the output of a real camera, the user of our rendering code should add to the noise-free gray level $\mathcal{I}(\mathbf{x})$ given by Eq. (9) a Gaussian signal-dependent noise of variance given by:

$$\operatorname{Var}(\mathcal{I}(\mathbf{x})) = g\mathcal{I}(\mathbf{x}) + \nu^2 - g\eta \tag{28}$$

where g is the gain, ν^2 is the readout noise level, and η is the offset value.

The Gaussian white noise model is, however, very common in papers on the assessment of DIC. It is backed by the possibility to use the Generalized Anscombe Transform [33] to normalize noise in Eq. (26). We thus give the user the opportunity to consider such a noise model when rendering speckle images.

4 Conclusion

The contribution of this paper is a new algorithm for rendering speckle images deformed by a displacement field prescribed by the user. It is based on the analysis of the image acquisition chain, from the infinite resolution speckle image seen as a realization of a Boolean model, to the digital image, taking into account the point spread function of the optical device through a Monte Carlo integration. Several parameters are carefully discussed to ensure that they are not responsible for additional biases, which is crucial in the context of the assessment of the metrological performance of DIC-based method in experimental mechanics. In particular, the sample size in Monte Carlo integration is set in such a way that the random fluctuation involved by this technique is mostly canceled out by quantization. It should also be noticed that no interpolation is required, interpolation being likely to give spurious gray values, as shown in Sect. 3.3.6. We have also discussed the effect of aliasing potentially caused by spatial sampling and the effect of rounding errors caused by floating point arithmetic. A continuation to this work would consist in the exploration of other speckle models than the Boolean model. Another potential perspective concerns the estimation of the parameters governing such models to render synthetic speckle images which would more precisely mimic real speckle patterns used in experimental mechanics, as mentioned in Sect. 3.4. Extending the proposed rendering algorithm to image pairs useful for stereo-DIC [4] or RGB-DIC [5] would also be interesting.

Acknowledgements F.S. is grateful to Antoine Fond (Magrit team, LORIA) for his help with the NVidia Titan X GPU.

Appendix A: Quantization Error for a Compound Probability Distribution

We assume that each real value inside a quantization box $[a, b]$ (with $a < b$) is assigned to a quantized value. In Monte Carlo estimation, sample means are distributed according to a normal distribution of mean m and standard deviation s. A quantization error occurs when the sample mean is not assigned to the same quantized value as m. If s is small with respect to $b - a$ and m is in the middle of the interval $[a, b]$, the probability of an error is small, while it is larger if m is close to a or b.

In order to estimate an overall error, we assume that the mean m is uniformly distributed over $[a, b]$. In other words, we assume that the sample mean X is a compound random variable distributed according to a Gaussian distribution of standard deviation s and mean m distributed according to a uniform distribution on interval $[a, b]$. We estimate the probability that X is not correctly quantized, i.e., that it falls outside the interval $[a, b]$. In other words, we would like to estimate the probability $\mathcal{E}(s, b - a) = 1 - \Pr(a \le X \le b)$.

We denote by g the standard normal probability distribution function (that is, $g(x) = e^{-x^2/2}/\sqrt{2\pi}$) and by G the associated cumulative distribution function. Marginalizing out m in X gives:

$$\Pr(a \le X \le b) = \frac{1}{b-a}\int_a^b \Pr(a \le X \le b \,|\, \mu)\, \mathrm{d}\mu \tag{29}$$

$$= \frac{1}{b-a}\int_a^b \left(G\left(\frac{b-\mu}{s}\right) - G\left(\frac{a-\mu}{s}\right)\right) \mathrm{d}\mu. \tag{30}$$

Now, for any two real values α and $\beta \neq 0$, an antiderivative of $G(\alpha + \beta x)$ is given by:

$$\int G(\alpha + \beta x) = \frac{1}{\beta}\left((\alpha + \beta x)G(\alpha + \beta x) + g(\alpha + \beta x)\right). \tag{31}$$

We calculate:

$$\frac{1}{b-a}\int_a^b G\left(\frac{b-\mu}{s}\right) \mathrm{d}\mu = G\left(\frac{b-a}{s}\right) + \frac{s}{b-a} g\left(\frac{b-a}{s}\right) - \frac{s}{\sqrt{2\pi}(b-a)} \tag{32}$$

and

$$\frac{1}{b-a}\int_a^b G\left(\frac{a-\mu}{s}\right) \mathrm{d}\mu = G\left(\frac{a-b}{s}\right) - \frac{s}{b-a} g\left(\frac{a-b}{s}\right) + \frac{s}{\sqrt{2\pi}(b-a)} \tag{33}$$

We eventually obtain, since $G(-x) = 1 - G(x)$ and $g(-x) = g(x)$:

$$\Pr(a \le X \le b) = 2G\left(\frac{b-a}{s}\right) - 1 + \frac{2s}{b-a}\left(g\left(\frac{b-a}{s}\right) - \frac{1}{\sqrt{2\pi}}\right) \quad (34)$$

We are interested in the quantization error when the random fluctuation of the Monte Carlo estimation is small, that is, when $s/(b-a)$ tends to 0.

Since $G(x) = 1 - g(x)/x + \mathcal{O}(g(x)/x)$ when $x \to +\infty$ (where $\mathcal{O}$ is Landau's "big-O"; this is a property of Mill's ratio [46, chapter 2]), we have proved the following proposition.

Proposition 1 *If a and b are two real numbers such that $a < b$, and X is a random variable distributed according to a compound Gaussian distribution of variance s^2 such that its mean is a random variable distributed according to a uniform distribution on $[a, b]$, the following asymptotic expansion holds:*

$$\Pr(a \le X \le b) =_{s \to 0} 1 - \frac{\sqrt{2}s}{\sqrt{\pi}(b-a)} + \mathcal{O}\left(se^{-(b-a)/(2s^2)}\right) \quad (35)$$

If $s/(b-a) << 1$, the latter expression gives an accurate estimate of the probability $\mathcal{E}$ of quantization error as:

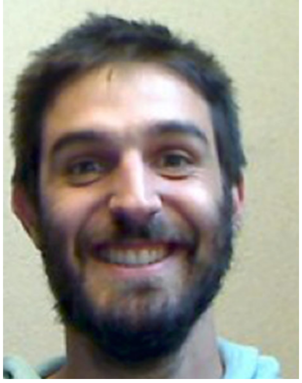
$$\mathcal{E}(s, b-a) = \frac{\sqrt{2}s}{\sqrt{\pi}(b-a)}. \quad (36)$$

References

1. Standard 1288, standard for characterization of image sensors and cameras, release 3.0. Tech. rep., European Machine Vision Association (EMVA) (2010)
2. Amiot, F., Bornert, M., Doumalin, P., Dupré, J.C., Fazzini, M., Orteu, J.J., Poilâne, C., Robert, L., Rotinat, R., Toussaint, E., Wattrisse, B., Wienin, J.: Assessment of digital image correlation measurement accuracy in the ultimate error regime: main results of a collaborative benchmark. Strain **49**(6), 483–496 (2013)
3. Baddeley, A.: Spatial point processes and their applications. In: Weil, W. (ed.) Stochastic Geometry: Lectures Given at the C.I.M.E. Summer School Held in Martina Franca, Italy, September 13–18, 2004, pp. 1–75 (2007)
4. Balcaen, R., Wittevrongel, L., Reu, P.L., Lava, P., Debruyne, D.: Stereo-DIC calibration and speckle image generator based on FE formulations. Exp. Mech. **57**(5), 703–718 (2017)
5. Baldi, A.: Digital image correlation and color cameras. Exp. Mech. (2017). https://doi.org/10.1007/s11340-017-0347-2
6. Barranger, Y., Doumalin, P., Dupré, J.C., Germaneau, A.: Strain measurement by digital image correlation: influence of two types of speckle patterns made from rigid or deformable marks. Strain **48**(5), 357–365 (2012)
7. Blaysat, B., Grédiac, M., Sur, F.: Effect of interpolation on noise propagation from images to DIC displacement maps. Int. J. Numer. Methods Eng. **108**(3), 213–232 (2016)
8. Blaysat, B., Grédiac, M., Sur, F.: On the propagation of camera sensor noise to displacement maps obtained by DIC—an experimental study. Exp. Mech. **56**(6), 919–944 (2016)
9. Blaysat, B., Grédiac, M., Sur, F.: Assessing the metrological performance of DIC applied on RGB images. In: Proceedings of the 2016 Annual Conference of the International Digital Imaging Correlation Society, Philadelphia (PA) USA (2017)
10. Bomarito, G., Hochhalter, J., Ruggles, T.: Development of optimal multiscale patterns for digital image correlation via local grayscale variation. Exp. Mech. (2017). https://doi.org/10.1007/s11340-017-0348-1
11. Bornert, M., Brémand, F., Doumalin, P., Dupré, J.C., Fazzini, M., Grédiac, M., Hild, F., Mistou, S., Molimard, J., Orteu, J.J., Robert, L., Surrel, Y., Vacher, P., Wattrisse, B.: Assessment of digital image correlation measurement errors: methodology and results. Exp. Mech. **49**(3), 353–370 (2009)
12. Bornert, M., Doumalin, P., Dupré, J.C., Poilâne, C., Robert, L., Toussaint, E., Wattrisse, B.: Short remarks about synthetic image generation in the context of sub-pixel accuracy of digital image correlation. In: Proceedings of the 15th International Conference on Experimental Mechanics (ICEM15), Porto, Portugal (2012)
13. Bornert, M., Doumalin, P., Dupré, J.C., Poilâne, C., Robert, L., Toussaint, E., Wattrisse, B.: Assessment of digital image correlation measurement accuracy in the ultimate error regime: improved models of systematic and random errors. Exp. Mech. (2017). https://doi.org/10.1007/s11340-017-0328-5
14. Bornert, M., Doumalin, P., Dupré, J.C., Poilâne, C., Robert, L., Toussaint, E., Wattrisse, B.: Shortcut in DIC error assessment induced by image inerpolation used for subpixel shifting. Opt. Lasers Eng. **91**, 124–133 (2017)
15. Briol, F.X., Oates, C., Girolami, M., Osborne, M., Sejdinovic, D.: Probabilistic integration: a role for statisticians in numerical analysis? Tech. Rep. arXiv:1512.00933, v5 (2016)
16. Cofaru, C., Philips, W., Paepegem, W.V.: Evaluation of digital image correlation techniques using realistic ground truth speckle images. Meas. Sci. Technol. **21**(5), 055,102/1–17 (2010)
17. Corless, R., Fillion, N.: A Graduate Introduction to Numerical Methods. Springer, Berlin (2014)
18. Doumalin, P., Bornert, M., Caldemaison, D.: Microextensometry by image correlation applied to micromechanical studies using the scanning electron microscopy. In: Proceedings of the International Conference on Advanced Technology in Experimental Mechanics, vol. I. The Japan Society of Mechanical Engineering, pp. 81–86 (1999)
19. Estrada, J., Franck, C.: Intuitive interface for the quantitative evaluation of speckle patterns for use in digital image and volume correlation techniques. J. Appl. Mech. **82**(9), 095,001-1–095,005-5 (2015)
20. Galerne, B.: Computation of the perimeter of measurable sets via their covariogram. Applications to random sets. Image Anal. Stereol. **30**(1), 39–51 (2011)
21. Galerne, B., Gousseau, Y.: The transparent dead leaves model. Adv. Appl. Probab. **44**(1), 1–20 (2012)
22. Gallego, M.A., Ibanez, M.V., Simó, A.: Parameter estimation in non-homogeneous Boolean models: an application to plant defense response. Image Anal. Stereol. **34**(1), 27–38 (2015)
23. Goldberg, D.: What every computer scientist should know about floating-point arithmetic. ACM Comput. Surv. **23**(1), 5–48 (1991)
24. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2000)
25. Healey, G., Kondepudy, R.: Radiometric CCD camera calibration and noise estimation. IEEE Trans. Pattern Anal. Mach. Intell. **16**(3), 267–276 (1994)

26. Hua, T., Xie, H., Wang, S., Hu, Z., Chen, P., Zhang, Q.: Evaluation of the quality of a speckle pattern in the digital image correlation method by mean subset fluctuation. Opt. Laser Technol. **43**(1), 9–13 (2011)
27. Koljonen, J., Alander, J.: Deformation image generation for testing a strain measurement algorithm. Opt. Eng. **47**(10), 107,202/1–13 (2008)
28. Lava, P., Cooreman, S., Coppieters, S., Strycker, M.D., Debruyne, D.: Assessment of measuring errors in DIC using deformation fields generated by plastic FEA. Opt. Lasers Eng. **47**(7–8), 747–753 (2009)
29. Lecompte, D., Smits, A., Bossuyt, S., Sol, H., Vantomme, J., Hemelrijck, D.V., Habraken, A.: Quality assessment of speckle patterns for digital image correlation. Opt. Lasers Eng. **44**(11), 1132–1145 (2006)
30. Lehoucq, R., Reu, P., Turner, D.: The effect of the ill-posed problem on quantitative error assessment in digital image correlation. Exp. Mech. (2017). https://doi.org/10.1007/s11340-017-0360-5
31. Mazzoleni, P., Matta, F., Zappa, E., Sutton, M., Cigada, A.: Gaussian pre-filtering for uncertainty minimization in digital image correlation using numerically-designed speckle patterns. Opt. Lasers Eng. **66**, 19–33 (2015)
32. Morel, J.M., Yu, G.: ASIFT: a new framework for fully affine invariant image comparison. SIAM J. Imaging Sci. **2**(2), 438–469 (2009)
33. Murthagh, F., Starck, J., Bijaoui, A.: Image restoration with noise suppression using a multiresolution support. Astron. Astrophys. **112**, 179–189 (1995)
34. Newson, A., Delon, J., Galerne, B.: A stochastic film-grain model for resolution-independent rendering. Comput. Graph. Forum **36**(8), 684–699 (2017)
35. Newson, A., Faraj, N., Delon, J., Galerne, B.: Analysis of a physically realistic film grain model, and a Gaussian film grain synthesis algorithm. In: Proceedings of the 6th Conference on Scale Space and Variational Methods in Computer Vision (SSVM), Kolding, Denmark (2017)
36. Newson, A., Faraj, N., Galerne, B., Delon, J.: Realistic film grain rendering. Image Process. Online (IPOL) **7**, 165–183 (2017)
37. Orteu, J.J., Garcia, D., Robert, L., Bugarin, F.: A speckle texture image generator. Proc. SPIE **6341**, 63,410H 1–6 (2006)
38. Pan, B., Xie, H.M., Xu, B.Q., Dai, F.L.: Performance of sub-pixel registration algorithms in digital image correlation. Meas. Sci. Technol **17**(6), 1615–1621 (2006)
39. Perlin, K.: An image synthesizer. SIGGRAPH Comput. Graph. **19**(3), 287–296 (1985)
40. Quenouille, M.: The evaluation of probabilities in a normal multivariate distribution, with special reference to the correlation ratio. Proc. Edinb. Math. Soc. **8**(3), 95–100 (1949)
41. Reu, P.L.: Experimental and numerical methods for exact subpixel shifting. Exp. Mech. **51**(4), 443–452 (2011)
42. Schneider, R., Weil, W.: Stochastic and Integral Geometry. Springer, Berlin (2008)
43. Schreier, H., Sutton, M.: Systematic errors in digital image correlation due to undermatched subset shape functions. Exp. Mech. **42**(3), 303–310 (2002)
44. Serra, J.: The Boolean model and random sets. Comput. Graph. Image Process. **12**(2), 99–126 (1980)
45. Serra, J.: Image Analysis and Mathematical Morphology. Academic Press, London (1982)
46. Small, C.: Expansions and Asymptotics for Statistics. Monographs on Statistics and Applied Probability, vol. 115. CRC Press, Boca Raton (2010)
47. Society for Experimental Mechanics: DIC challenge. https://sem.org/dic-challenge/
48. Stoyan, D., Kendall, W., Mecke, J., Kendall, D.: Stochastic Geometry and Its Applications. Wiley, New York (1987)
49. Su, Y., Zhang, Q., Gao, Z.: Statistical model for speckle pattern optimization. Opt. Express **25**(24), 30259–30275 (2017)
50. Sutton, M., Orteu, J.J., Schreier, H.: Image Correlation for Shape, Motion and Deformation Measurements. Springer, Berlin (2009)
51. Triconnet, K., Derrien, K., Hild, F., Baptiste, D.: Parameter choice for optimized digital image correlation. Opt. Lasers Eng. **47**(6), 728–737 (2009)
52. Yu, L., Pan, B.: The errors in digital image correlation due to overmatched shape functions. Meas. Sci. Technol. **26**(4), 045–202 (2015)

Frédéric Sur received a Ph.D. from Université Paris-Dauphine in 2004. He has been an associate professor at Université de Lorraine (Nancy, France) from 2006. His current research interests include image analysis, image processing, and applications to computational photomechanics.



Benoît Blaysat after a Ph.D. in computational mechanics (defended in 2011 at the LMT-Cachan, France), Benoît Blaysat got a post-doctoral fellowship in the Netherlands (TU/e, 2011–2012) focused on the use of full field measurement techniques in experimental mechanics. In Sept. 2013, he joined the Université Clermont-Auvergne (France) as an assistant professor. He is a member of the Institut Pascal laboratory. His main research interest is the development and use of full-field measurement techniques in experimental solid mechanics.

Michel Grédiac is a professor at the Université Clermont-Auvergne, Clermont-Ferrand (France). He is a member of the Institut Pascal laboratory. His main research interest is the development and use of full-field measurement techniques in experimental solid mechanics.