Experimental Mechanics
DOI 10.1007/s11340-015-0009-1

Ncorr: Open-Source 2D Digital Image Correlation Matlab Software

J. Blaber[1] · **B. Adair**[1] · **A. Antoniou**[1]

Received: 14 August 2014 / Accepted: 9 March 2015

Abstract Digital Image Correlation (DIC) is an important and widely used non-contact technique for measuring material deformation. Considerable progress has been made in recent decades in both developing new experimental DIC techniques and in enhancing the performance of the relevant computational algorithms. Despite this progress, there is a distinct lack of a freely available, high-quality, flexible DIC software. This paper documents a new DIC software package Ncorr that is meant to fill that crucial gap. Ncorr is an open-source subset-based 2D DIC package that amalgamates modern DIC algorithms proposed in the literature with additional enhancements. Several applications of Ncorr that both validate it and showcase its capabilities are discussed.

Keywords Digital image correlation · Large deformation · Complex ROI · Nickel superalloy · Crack

Introduction

Digital Image Correlation (DIC) is a robust non-contact technique for measuring material deformation [1–3]. DIC uses image registration algorithms to track the relative displacements of material points between a reference (typically, the undeformed) image and a current (typically, the deformed) image [4, 5]. The scale agnostic nature of DIC allows for the study of deformation at different length scales from meters to the nanoscale [6–11], as long as the material can be properly patterned and imaged. The behavior of diverse systems has been studied such as biological materials [12–15], metal alloys [16–18], shape memory alloys [19, 20], porous metals [21–23], polymers [24] and polymer foams [25]. Each material system presents challenges that may require modification of standard DIC algorithms, such as highly localized deformation, large strains, etc.

In recent years, several improvements to the core computation algorithms have been proposed and implemented [26–31]. However, there appears to be a distinct lack of a modern, adaptable, and accessible DIC code (although older packages exist on the Matlab File Exchange [32, 33]). In order to accelerate innovation and to introduce DIC to a wide community of researchers, we have developed an open-source, freely available 2D subset-based digital image correlation software package (Ncorr) that amalgamates modern algorithms proposed in the literature [28, 31, 34–37], as well as many additional improvements and changes. Ncorr is meant to be a well-documented flexible code that users can freely adapt to suit their needs. The package is implemented in MATLAB (http://www.mathworks.com/products/matlab/), a widely used general purpose numerical computing environment for engineers. For greater efficiency, compute intensive algorithms are written in C++ and incorporated into MATLAB through the MEX interface (http://www.mathworks.com/help/matlab/share-mex-files.html). A high quality Graphical User Interface (GUI) can assist in the analysis.

The purpose of this paper is to document the algorithms implemented in Ncorr, as well as the validation and verification tests. We also demonstrate application of Ncorr to several

Electronic supplementary material The online version of this article (doi:10.1007/s11340-015-0009-1) contains supplementary material, which is available to authorized users.

✉ A. Antoniou
antonia.antoniou@me.gatech.edu

[1] The Woodruff School of Mechanical Engineering, 801 Ferst Drive, Atlanta, GA 30332, USA

Published online: 31 March 2015

common problems in experimental mechanics that showcase some of the code capabilities.

The significance of DIC for experimental mechanics community is exemplified by the recent Society of Experimental Mechanics (SEM) DIC challenge (http://www.sem.org/dic-challenge/). The aim of the challenge is to cross validate existing DIC codes and to document their performance on a well documented set of standard tests. The analysis of several examples from the DIC challenge using Ncorr is discussed in this paper.

The paper is organized as follows: 1) description of fundamental DIC algorithms used in Ncorr, 2) extensions and improvements of the code specific to Ncorr, 3) validation and 4) representative applications.

Core DIC Algorithm Description

Subset Deformation

In subset-based DIC algorithms, the reference image is partitioned into smaller regions referred to as subsets or subwindows. The deformation is assumed to be homogeneous inside each subset, and the deformed subsets are then tracked in the current image. In Ncorr, subsets are initially a contiguous circular group of points that are on integer pixel locations in the reference configuration. The transformation of the coordinates of these points from the reference to the current configuration is constrained to a linear, first order transformation

$$\begin{aligned}\widetilde{x}_{cur_i} &= x_{ref_i} + u_{rc} + \frac{\partial u}{\partial x_{rc}}\left(x_{ref_i} - x_{ref_c}\right) + \frac{\partial u}{\partial y_{rc}}\left(y_{ref_j} - y_{ref_c}\right) \\ \widetilde{y}_{cur_j} &= y_{ref_j} + v_{rc} + \frac{\partial v}{\partial x_{rc}}\left(x_{ref_i} - x_{ref_c}\right) + \frac{\partial v}{\partial y_{rc}}\left(y_{ref_j} - y_{ref_c}\right)(i,j)\in S\end{aligned} \tag{1}$$

$$\mathbf{p} = \left\{ u \quad v \quad \frac{\partial u}{\partial x} \quad \frac{\partial u}{\partial y} \quad \frac{\partial v}{\partial x} \quad \frac{\partial v}{\partial y} \right\}^T \tag{2}$$

Here x_{ref_i} and y_{ref_j} are the x and y coordinates of an initial reference subset point, x_{ref_c} and y_{ref_c} are the coordinates of the center of the initial reference subset, and $\widetilde{x}_{cur_i}$ and $\widetilde{y}_{cur_j}$ are the coordinates of a current subset point. The deformation is parameterized by displacements u and v and their derivatives, all of which are constant for a given subset. Equation (2) defines a generalized deformation vector $\boldsymbol{p}$. The indices (i,j) are used for the relative location of the points with respect to the center of the subset, as well as to establish correspondences between subset points in the current and reference configuration and S is a set which contains all of the subset points. The subscript "rc" used in equation (1) is meant to signify that the transformation is from the reference to current coordinate system. Equation (1) can also be written in matrix form

$$\xi_{\mathrm{ref_c}} + w(\Delta\xi_{ref};\mathbf{p}_{\mathrm{rc}}) = \begin{Bmatrix} x^{\mathrm{T}}_{\mathrm{ref_c}} \\ y^{\mathrm{T}}_{\mathrm{ref_c}} \\ 1 \end{Bmatrix} + \begin{bmatrix} 1+\frac{\mathrm{du}}{\mathrm{dx_{rc}}} & \frac{\mathrm{du}}{\mathrm{dy_{rc}}} & \mathrm{u_{rc}} \\ \frac{\mathrm{dv}}{\mathrm{dx_{rc}}} & 1+\frac{\mathrm{dv}}{\mathrm{dy_{rc}}} & \mathrm{v_{rc}} \\ 0 & 0 & 1 \end{bmatrix} * \begin{Bmatrix} \Delta x^{\mathrm{T}}_{\mathrm{ref}} \\ \Delta y^{\mathrm{T}}_{\mathrm{ref}} \\ 1 \end{Bmatrix} \tag{3}$$

where ξ is an augmented vector containing the x and y coordinates of subset points, Δx and Δy are the distances between subset points and the center of the subset, and w is a so-called warp function.

For purposes of computational efficiency (more specifically, to accommodate the inverse compositional method described later), we also allow the reference subset to deform within the reference configuration as follows:

$$\begin{aligned}\widetilde{x}_{ref_i} &= x_{ref_i} + u_{rr} + \frac{\partial u}{\partial x_{rr}}\left(x_{ref_i} - x_{ref_c}\right) + \frac{\partial u}{\partial y_{rr}}\left(y_{ref_j} - y_{ref_c}\right) \\ \widetilde{y}_{ref_j} &= y_{ref_j} + v_{rr} + \frac{\partial v}{\partial x_{rr}}\left(x_{ref_i} - x_{ref_c}\right) + \frac{\partial v}{\partial y_{rr}}\left(y_{ref_i} - y_{ref_c}\right)(i,j)\in S\end{aligned} \tag{4}$$

where $\widetilde{x}_{ref_i}$ and $\widetilde{y}_{ref_j}$ are the x and y coordinates of a deformed reference subset point. The "rr" subscript in equation (4) is meant to signify that the transformation is between two different coordinate systems in the reference image.

Correlation Criteria

In order to find the deformation of a subset, DIC algorithms find the extremum of a correlation (cost) function. Figure 1 shows a schematic of this process. In Ncorr, two different correlation criteria are used to find the initial guess and its subsequent refinement. The initial guess is found by computing at integer locations the normalized cross correlation (NCC) (see Fig. 2)

$$C_{cc} = \frac{\sum_{(i,j)\in S}\left(f\left(\widetilde{x}_{ref_i},\widetilde{y}_{ref_j}\right)-f_m\right)\left(g\left(\widetilde{x}_{cur_i},\widetilde{y}_{cur_j}\right)-g_m\right)}{\sqrt{\sum_{(i,j)\in S}\left[f\left(\widetilde{x}_{ref_i},\widetilde{y}_{ref_j}\right)-f_m\right]^2\sum_{(i,j)\in S}\left[g\left(\widetilde{x}_{cur_i},\widetilde{y}_{cur_j}\right)-g_m\right]^2}} \tag{5}$$

Here f and g are respectively the reference and current image grayscale intensity functions at a specified location *(x, y)*. Functions f_m and g_m correspond to the mean grayscale values of the reference and current subset

$$f_m = \frac{\sum_{(i,j)\in S} f\left(\widetilde{x}_{ref_i},\widetilde{y}_{ref_j}\right)}{n(S)} \tag{6}$$

$$g_m = \frac{\sum_{(i,j)\in S} g\left(\widetilde{x}_{cur_i},\widetilde{y}_{cur_j}\right)}{n(S)} \tag{7}$$

where *n(S)* is the number of data points in subset *S*. The

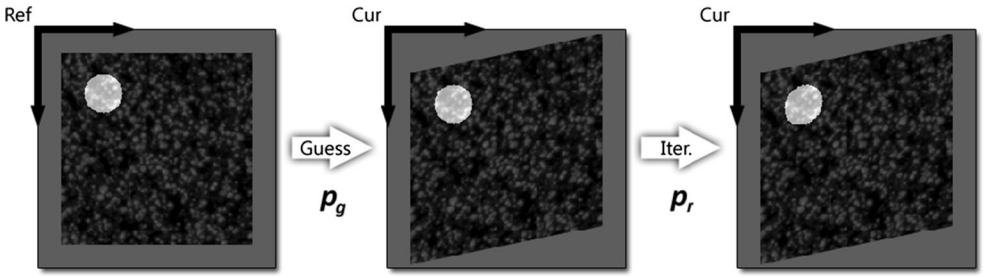

Fig. 1 The basic steps of DIC algorithm implemented in Ncorr. First, the initial guess for the displacement $p_g=\{u^{(g)},v^{(g)},0,0,0,0\}^T$ is found. This guess is used as the initial input to the iterative optimization scheme, which finds a refined solution $p_r=\{u,v,\partial u/\partial x,\partial u/\partial y,\partial v/\partial x,\partial v/\partial y\}^T$ allowing for subwindow deformation

initial guess thus yields u and v with integer (pixel) accuracy. The next step uses a nonlinear optimizer to refine these results with sub-pixel resolution by finding the minimum of

$$C_{LS}=\sum_{(i,j)\in S}\left[\frac{f\left(\tilde{x}_{ref_i},\tilde{y}_{ref_j}\right)-f_m}{\sqrt{\sum_{(i,j)\in S}\left[g\left(\tilde{x}_{ref_i},\tilde{y}_{ref_j}\right)-f_m\right]^2}}-\frac{g\left(\tilde{x}_{cur_i},\tilde{y}_{cur_j}\right)-g_m}{\sqrt{\sum_{(i,j)\in S}\left[g\left(\tilde{x}_{cur_i},\tilde{y}_{cur_j}\right)-g_m\right]^2}}\right]^2 \tag{8}$$

Correlation criteria C_{CC} and C_{LS} are directly related [38].

Non-Linear Optimization Scheme

Ncorr uses the Inverse Compositional Gauss-Newton (IC-GN) method as the nonlinear optimizer [39, 40]. IC-GN is a computationally efficient iterative method that seeks to find an optimal value of $\boldsymbol{p}_{rc}$ (denoted as $\boldsymbol{p}^*_{rc}$) that minimizes C_{LS} when $\boldsymbol{p}_{rr}=0$. In each iteration, IC-GN method finds a small deformation $\Delta\boldsymbol{p}$ ($\boldsymbol{p}_{rr}$ in equation (4)) of the initial reference subset that best matches the deformed reference subset described by $\boldsymbol{p}_{rc}$. The resulting $\boldsymbol{p}_{rr}$ is then used to update $\boldsymbol{p}_{rc}$ (see Fig. 3). The compact form of C_{LS} for the IC-GN iterations is:

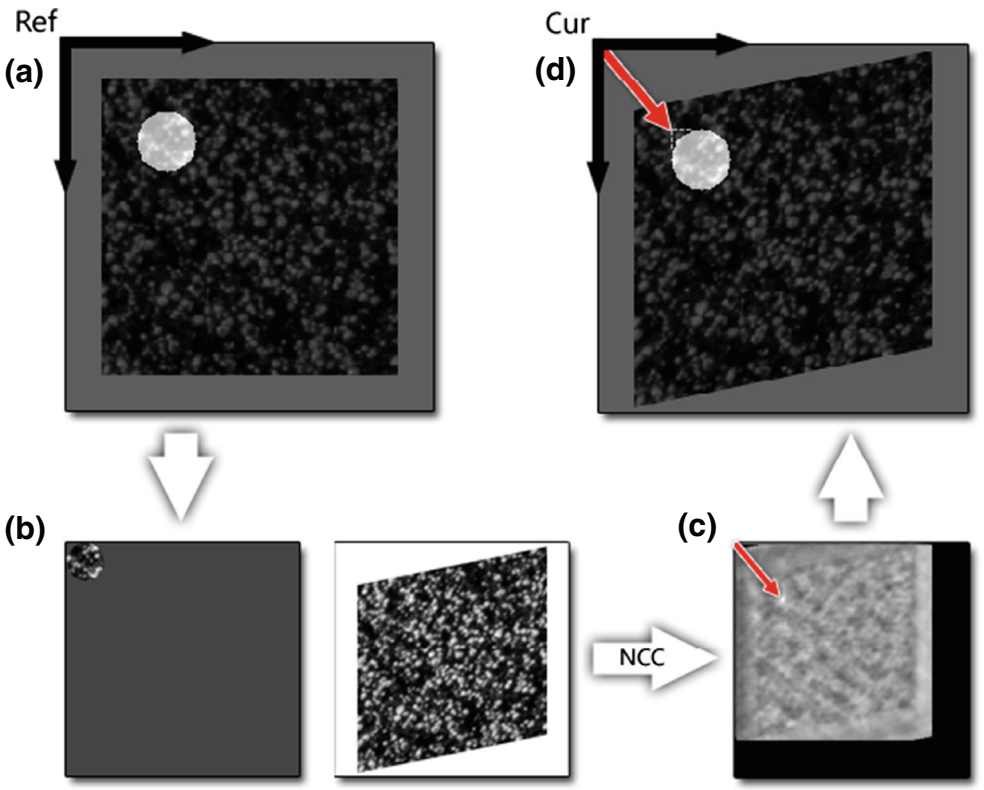

Fig. 2 The procedure for finding the initial guess. A reference subset (**a**) is padded to the full-size (**b**) and is convolved with the current image to find the normalized cross correlation (**c**). The output is an array of correlation coefficient values. The location of the subset is recovered with respect to the current configuration by locating the maximum correlation coefficient (**d**)

$$C_{LS}(\Delta p)=\sum\left[\frac{f\left(\xi_{\mathrm{ref_c}}+w(\Delta\xi_{ref};\Delta p)\right)-f_m}{\sqrt{\sum\left[f\left(\xi_{\mathrm{ref_c}}+w(\Delta\xi_{ref};\Delta p)\right)-f_m\right]^2}}-\frac{g\left(\xi_{\mathrm{ref_c}}+w(\Delta\xi_{ref};p_{old})\right)-g_m}{\sqrt{\sum\left[g\left(\xi_{\mathrm{ref_c}}+w(\Delta\xi_{ref};p_{old})\right)-g_m\right]^2}}\right]^2 \tag{9}$$

where $\mathbf{p}_{\mathrm{old}}$ is the value of $\mathbf{p}_{\mathrm{rc}}$ found from the previous iteration. In order to find the minimum of C_{LS}, we employ an iterative procedure based on the Taylor series expansion

$$\nabla\nabla C_{LS}(\mathbf{0})\Delta\boldsymbol{p}=-\nabla C_{LS}(\mathbf{0}) \tag{10}$$

where $\nabla C_{\mathrm{LS}}(\mathbf{0})$ is the gradient of C_{LS} at $\mathbf{p}=\mathbf{0}$ and $\nabla\nabla C_{\mathrm{LS}}(\mathbf{0})$ is the hessian of C_{LS} at $\mathbf{p}=\mathbf{0}$. The explicit form of these quantities is shown in Appendix A1. Note that the computation requires an interpolation scheme. Ncorr uses Biquintic B-splines, as described in Appendix A2. In each iteration, $\Delta\mathbf{p}$ is solved using Cholesky decomposition since the initial guess ensures that parameters are in the vicinity of a minimum, where the hessian should be symmetric positive definite. If the Cholesky decomposition fails, then the hessian is not positive definite and the point is rejected. The next approximation to $\mathbf{p}^*_{\mathbf{rc}}$ is found by composing the old value with the inverse of $\Delta\mathbf{p}$, which is justified so long as the displacements in $\Delta\mathbf{p}$ are small in magnitude. A graphical representation of the update is shown in Fig. 3. This update can be written in matrix form as follows

$$\begin{bmatrix}1+\frac{du}{dx_{new}} & \frac{du}{dy_{new}} & u_{new}\\ \frac{dv}{dx_{new}} & 1+\frac{dv}{dy_{new}} & v_{new}\\ 0 & 0 & 1\end{bmatrix}=\begin{bmatrix}1+\frac{du}{dx_{old}} & \frac{du}{dy_{old}} & u_{old}\\ \frac{dv}{dx_{old}} & 1+\frac{dv}{dy_{old}} & v_{old}\\ 0 & 0 & 1\end{bmatrix}*\begin{bmatrix}1+\Delta\frac{du}{dx} & \Delta\frac{du}{dy} & \Delta u\\ \Delta\frac{dv}{dx} & 1+\Delta\frac{dv}{dy} & \Delta v\\ 0 & 0 & 1\end{bmatrix}^{-1} \tag{11}$$

$$w(\Delta\xi_{ref};\mathbf{p}_{\mathrm{new}})=w\left(w(\Delta\xi_{ref};\Delta\mathbf{p})^{-1};\mathbf{p}_{\mathbf{old}}\right) \tag{12}$$

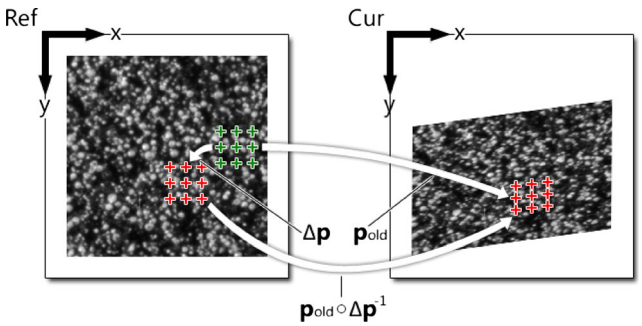

Fig. 3 A graphical representation of the inverse compositional update

Here $\mathbf{p}_{\mathrm{old}}$ is set to $\mathbf{p}_{\mathrm{new}}$ at the beginning of every iteration. The inverse compositional method gets its name from how $\mathbf{p}_{\mathrm{old}}$ is updated in equations (11–12). The crucial advantage of the method is that the hessian is computed at $\xi_{\mathrm{ref_c}}+w\left(\Delta\xi_{ref};\mathbf{0}\right)$ and does not change during iterations, i.e. it only needs to be computed once. This results in large computational savings [39].

A summary of the inverse compositional method is shown below (the steps follow from Ref. [39] but have been adapted for DIC):

Precomputation for entire DIC Analysis:

1) (*Optional*) Precompute $[QK][c]_{(x_f-2:x_f+3,y_f-2:y_f+3)}[\mathrm{QK}]^T$ ([QK] is defined in equation (34) and [c] in equation (35)) for the entire current image.
2) Evaluate $\frac{\partial}{\partial\widetilde{x}_{ref_i}}f\left(\widetilde{x}_{ref_i},\widetilde{y}_{ref_j}\right)$ and $\frac{\partial}{\partial\widetilde{y}_{ref_j}}f\left(\widetilde{x}_{ref_i},\widetilde{y}_{ref_j}\right)$ for the entire reference image using equation (37) and equation (38).

Precomputation per subset:

3) Compute "steepest descent images", equation (25).
4) Compute the GN-Hessian in equation (24).
5) Set initial $\mathbf{p}_{\mathrm{old}}$ to initial guess from NCC or neighboring deformation data

Computation per iteration per subset:

6) Compute warped final current subset $g\left(\xi_{\mathrm{ref_c}}+w(\Delta\xi_{ref};\boldsymbol{p_{old}})\right)$ using equation (33)
7) Compute gradient, $\nabla C_{LS}(\mathbf{0})$, using equation (21)
8) Compute $\Delta\mathbf{p}$ using equation (10) with Cholesky decomposition
9) Update $\mathbf{p}_{\mathrm{old}}$ using equation (11)
10) Exit iteration when norm of $\Delta\mathbf{p}$ is small.

Reliability Guided DIC

The steps described previously allow for the calculation of displacement data of a single material point located at the center of a reference subset. In order to obtain displacement values for all subsets, Ncorr uses the Reliability Guided (RG-DIC) method [41]. For completeness, a brief summary of the method is given below, although the reader is directed to Ref. [41] for an in depth description.

The overall method is shown in Figs. 4 and 5. The computation begins by selecting a seed point. This is the location of the center of the initial reference subset used in the analysis and is shown as the green cross with "Seed" in Fig. 5. This point is the only one which uses NCC to obtain an initial guess. The rest of the subsets will use neighboring information as an initial guess.

After calculating the corresponding deformation parameters and C_{LS} for the seed point, it is "inactivated," which is done by forming a logical mask initialized to false; analyzed points are set to true. A queue is formed (as a heap data structure) and then C_{LS}, the six deformation parameters, and the location of the center of the subset, $\left(x_{ref_c}, y_{ref_c}\right)$, are stored in the queue, which is shown as the 2nd step in the flow diagram of Fig. 4. The program enters a while loop and at each iteration the top entry (this entry will have the lowest C_{LS} since the queue is a heap) is popped from the queue. The data for the queue point is then added to the data plots. Next, the four surrounding material points, if they are still active or within the region of interest (ROI), are analyzed using the displacement data from the queue point as the initial guess for the nonlinear optimization scheme. The process is repeated until the queue is empty; at this stage all contiguous points of interest will be analyzed. The aforementioned process is summarized in the flow diagram of Fig. 4.

The benefits of this process are two-fold. First of all, it is robust in that bad data points (i.e. data points with high C_{LS}) are processed last, which prevents this data from being used as the initial guess for neighboring points. In addition, it is computationally efficient because only the seedpoint needs to use NCC (which is computationally expensive) to obtain an initial guess. Moreover, the initial guess provided by adjacent subset is typically quite good (as long as the subset spacing is relatively small) because displacement fields are, in general, relatively smooth. One negative aspect of this algorithm is that it must be done serially, although in Ncorr this has been circumvented by partitioning the ROI into spatial domains that are calculated in parallel. This process is described in more detail in "Multithreaded RG-DIC" section.

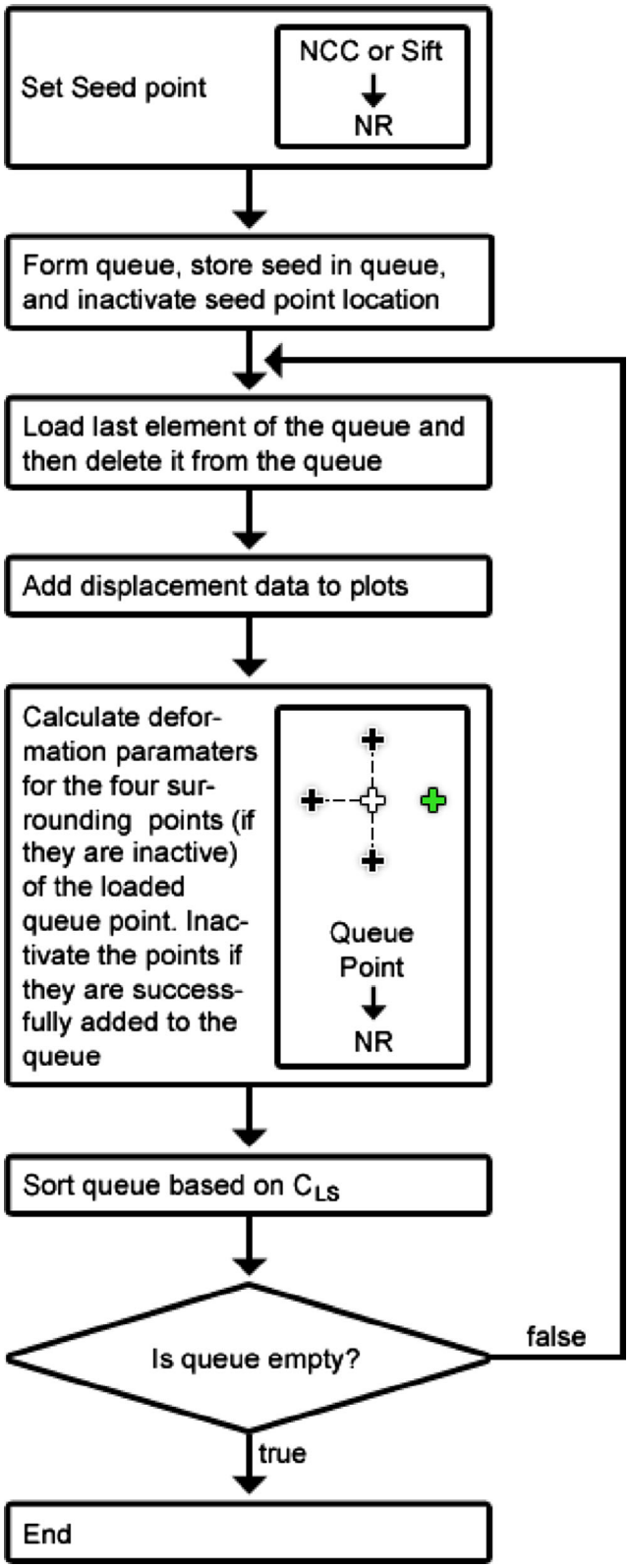

Fig. 4 Flow chart of the RG-DIC algorithm. In the neighbor calculation phase, the central white cross is the material point loaded from the queue. The three *black crosses* are unanalyzed points which are added to the queue. The *green cross* has already been analyzed and is skipped

Computation of Strains

Strains are more difficult to resolve than the displacement fields because strains involve differentiation, which is sensitive to noise. This means any noise in the displacement field will magnify errors in the strain fields. Ncorr uses the Green-Lagrangian strains, which are obtained by using the four displacement gradients as follows

$$E_{xx} = \frac{1}{2}\left(2\frac{\partial u}{\partial x} + \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2\right) \tag{13}$$

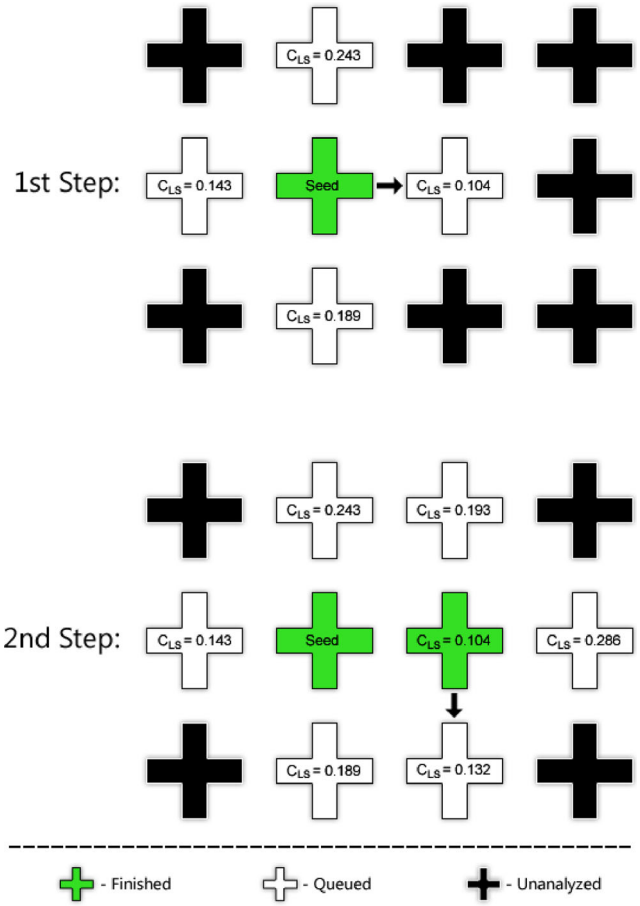

Fig. 5 The approach used by RG-DIC algorithm when determining deformation parameters for a grid within a region of interest. The direction proceeds in the direction of the lowest C_{LS} value

$$E_{xy} = \frac{1}{2}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} + \frac{\partial u}{\partial x}\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\frac{\partial v}{\partial y}\right) \tag{14}$$

$$E_{yy} = \frac{1}{2}\left(2\frac{\partial v}{\partial y} + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2\right) \tag{15}$$

The displacement gradients are directly obtained through the IC-GN scheme. But these values are noisy, and thus must be "smoothed" in some way before calculating the strain fields. Instead, Ncorr uses the strain window algorithm proposed in Ref. [35] to calculate the displacement gradients and the subsequent Green-Lagrangian strains. A least squares plane fit on a subset of displacement data is used to find the plane parameters in equations (13–15) and thus the displacement gradient data from the RG-DIC algorithm is not utilized

$$u_{plane}(x,y) = a_{u,plane} + \left(\frac{\partial u}{\partial x_{plane}}\right)x + \left(\frac{\partial u}{\partial y_{plane}}\right)y \tag{16}$$

$$v_{plane}(x,y) = a_{v,plane} + \left(\frac{\partial v}{\partial x_{plane}}\right)x + \left(\frac{\partial v}{\partial y_{plane}}\right)y \tag{17}$$

The strains are found as a solution of an over-constrained system of equations [42]

$$\begin{bmatrix} 1 & x_{ref_{first\ i}} - x_{ref_c} & y_{ref_{first\ j}} - y_{ref_c} \\ \vdots & \vdots & \vdots \\ 1 & x_{ref_{last\ i}} = x_{ref_c} & y_{ref_{last\ j}} - y_{ref_c} \end{bmatrix} \begin{Bmatrix} a_{u,plane} \\ \left(\frac{\partial u}{\partial x_{plane}}\right) \\ \left(\frac{\partial u}{\partial y_{plane}}\right) \end{Bmatrix} = \begin{Bmatrix} u^*_{rc}\left(x_{ref_{first\ i}}, y_{ref_{first\ j}}\right) \\ \vdots \\ u^*_{rc}\left(x_{ref_{last\ i}}, y_{ref_{last\ j}}\right) \end{Bmatrix} \tag{18}$$

The size of the subset window for computing the strains can be independently controlled. Once these parameters are solved, they can be used in equations (13–15) to determine E_{xx}, E_{xy}, and E_{yy}. The process is applied to the entire displacement field in order to obtain a corresponding strain field.

New Algorithms Implemented in Ncorr

New algorithms implemented in Ncorr are: 1) a multithreaded version of RG-DIC, 2) a method for large strain calculation by automatically updating the ROI and adding displacements, and 3) an Eulerian to Lagrangian conversion algorithm to analyze discontinuous displacement fields.

Multithreaded RG-DIC

Ncorr extends RG-DIC to multithreading by forming a "thread-diagram" (an integer array) based on seed point locations within a 4-way connected region in the ROI. This idea is demonstrated in Fig. 6. It starts by growing subregions around each seed, one point at a time per iteration, until the entire ROI has been segmented. This ensures the region around each seed is contiguous, and also divides the work relatively evenly. An example thread diagram is shown on the right side of Fig. 6. It is an array that is same size as the original ROI, but has values which correspond to each thread. For example, the region marked "thread 0" on Fig. 6 contains zeros within that region, and for "thread 1", the array has ones, etc., while the black region outside the ROI contains negative ones (an impossible value for a thread). This idea is well suited for multithreading because it scales well, in that as the limit of the number of seeds approach the number of points within the region of interest, it simply approaches the older methods of DIC, wherein every subset's initial guess is found through NCC.

ROI Updating for Large Deformations

A challenging problem in DIC involves material undergoing large deformations. The deformed sample may change so dramatically that a good correlation of DIC subsets between

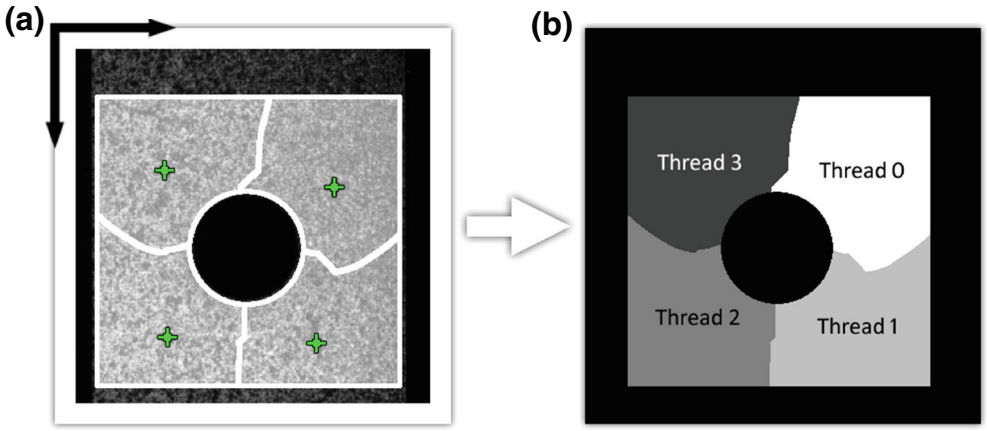

Fig. 6 Ncorr implementation of multithreaded RG-DIC. (**a**) Four seeds are placed within an ROI, which is then partitioned based on seed placement. (**b**) Each partition is then assigned a thread through a "thread diagram"

current and reference image will not be possible. At the same time, there is significant interest in applying DIC to materials that can undergo large deformations such as soft materials, polymers, elastomers as well as biological materials. A strategy to monitor deformation of such materials involves a well patterned sample and a dataset of images at intermediate stages of the deformation so that the deformation within two adjacent images is not large enough to cause problems. Ncorr has been properly modified to create composite displacement maps that can subsequently be used to obtain the strain field. An important part of this process involves updating the ROI. The method implemented in Ncorr works by forming a boundary around the ROI. This boundary is then updated based on displacement values obtained through the last calculated displacement field for the last reference image. This happens by sampling the last calculated displacement field at the values at the boundary. These displacement values are then added to the boundary coordinates. The boundary coordinates are treated as a polygon, and are used through a polygon fill algorithm [43] to update the ROI. Updates occur either when the seeds placed on the original reference image either exceed certain cutoffs (i.e. correlation coefficient or the number of allowable iterations for convergence) or through a preset number, known as a "leapfrog" parameter. The boundary updating procedure is demonstrated in Fig. 7.

It's important to note that most of the compounding error will occur for datapoints near the boundary of the ROI. This is because the displacement fields are interpolated with biquintic B-splines, so the fields actually need to be extrapolated first before obtaining the B-spline coefficients for datapoints near the border. Therefore, interpolated values near the border will be the most affected. However, one of the positive aspects of this update scheme is that because the ROI itself is updated when the reference image is updated (unlike the approach of Ref. [27]), the subset deformations are not perpetuated. This means that although this method might result in worse performance for boundary displacement points, it's possible to analyze samples undergoing large deformations, provided enough intermediate pictures are provided and the pattern on the updated reference images is sufficient enough to allow for the tracking algorithms to perform correctly.

Eulerian to Lagrangian Conversion

This technique is used to analyze discontinuous displacement fields in Ncorr. It relies on the fact that discontinuities (i.e. cracks) are clearly visible in the current image. The idea is to create the ROI for the current image (rather than the reference image), perform DIC, and then use an Eulerian to Lagranian algorithm to convert the displacements back to the Lagrangian perspective.

The ROI is updated through a similar technique used in the large strain calculation algorithm. Once the ROI is updated (shown in Fig. 8), every point in the updated ROI is analyzed

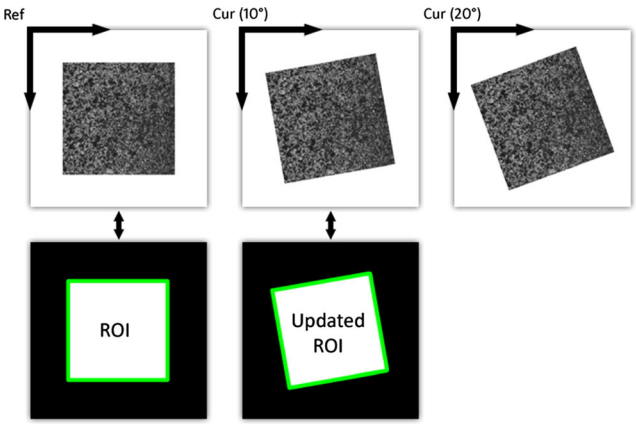

Fig. 7 A graphical example of ROI updating and displacement "adding" scheme. Here, there is a single update between the current image rotated 10° and the current image rotated 20°. When the reference image needs to update, the boundary (highlighted in *green*) is used to sample displacement values (shown in the *bottom row*) in order to update the boundary. This boundary is then used to form the updated ROI through a polygon fill routine. The displacements are then "added" by sampling displacement values using the original reference image and then finding their corresponding position in subsequent displacement fields (note that this requires interpolation). The end result is shown on the *bottom right*

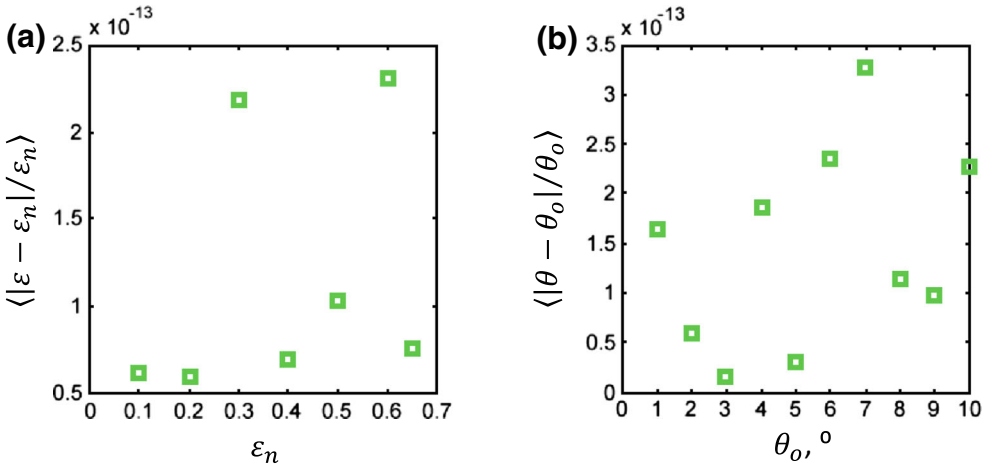

Fig. 8 The error in strain measurements in two verification tests. (**a**) Uniform strain applied at an angle of θ=30° with respect to the horizontal axis. Here ε_n is the prescribed strain and ε is the measured strain. Angle brackets denote the average over ROI. (**b**) Rigid body rotation. Here θ_o is the prescribed angle and θ is the measured angle. Angle brackets denote the average over ROI

by first determining which integer point in the current displacement field maps closest to that point. This is fed as an initial guess to a nonlinear optimizer to determine the sub pixel displacements using biquintic B-spline interpolation on the displacement field. This initial point also acts as a seed and initiates a queue. The algorithm then proceeds to use neighboring points as the initial guess to the nonlinear optimizer in much the same way as the RG-DIC algorithm described in the "Core DIC Algorithm Description" section. The routine finishes when all the points in the updated ROI have been analyzed, which results in a displacement field with respect to the Lagrangian perspective.

Validation and Verification

The code was verified and validated using a variety of tests. The first set of tests involved idealized simulated images, where results could be verified to numerical precision. Further tests involved analysis of the Society for Experimental Mechanics's (SEM) 2D-DIC challenge simulated datasets. The DIC challenge provides a number of tests that simulate many of the problems and issues encountered in applications, such as noisy images and patterns of limited resolution and contrast. The full analysis of simulated SEM 2D-DIC images can be found in the Supplementary Material section accompanying the online version of this article. The analysis of one representative example, Sample 14 from the DIC challenge, is shown in this section.

Uniaxial Strain State and Rigid Body Rotation

Numerically simulated images were used to demonstrate the correctness of the implementation of the algorithms. The simulated images were resampled using biquintic B-splines in order to match the interpolation scheme used by Ncorr. This is a stringent test of parts of the algorithm, since the results should be "exact." The current image was set as the undeformed image (taken previously with a generic speckle pattern). Then, the inverse of the desired transformation is applied to the current image using biquintic B-splines in order to obtain the reference image. If the undeformed image is set as the reference image and then a forward transformation is applied to it, the results will not be "exact," even if biquintic B-splines are used, since in the forward transformation the reference image is being interpolated. The images were saved as .mat files since any standard image formats (jpg, png, tif, etc.…) use either unsigned integers or a form of data compression which will alter the data.

The imposed known deformation for the image is a uniaxial stretch of increasing magnitude, ε_n along a 30° angle with respect to the horizontal direction. A DIC subset radius r= 15 pixels and a strain window of 15 pixels were used in the analysis of images with a size of 600×600 pixels. The imposed strain ε_n ranged from 0.1 to 0.65 since beyond 0.65, the normalized cross correlation algorithm failed to provide the initial guess [44]. Through a simple strain transformation, the strains ε_{xx}, ε_{xy}, and ε_{yy} were obtained. These strains were applied by determining the displacement gradients from equations (13–15) through a nonlinear solver. The transformation was formed using an augmented matrix from equation (11) and then the inverse transformation was applied to the current image to obtain the reference image. The results from the analysis are shown in Fig. 8(a) where the average absolute value of the error in the calculated ε_n was obtained from a transformation of the three Green-Lagrangian strains over the ROI. We note that the initial guess in Ncorr is obtained by computing NCC at integer location assuming that the subset deformation does not involve strains. This assumption is clearly violated when strains are large, which is the reason for inability of the code to analyze strains larger than 0.65 in this example. In practical applications, this limitation is overcome by using intermediate images to propagate the initial guess. The intermediate analysis was intentionally not used here,

even though it is implemented in Ncorr (as described in "ROI Updating for Large Deformations" section).

A similar analysis was performed for images undergoing rigid body rotation (RBR) and their analyzed Ncorr result was compared to the prescribed value. The synthesized rotated image was formed using the same set as the uniaxial stretch and the analysis was done with the same subset and strain radius sizes as the uniaxial strain. The rotation images were generated up until a rotation of about 10°, because after that, the normalized cross correlation algorithm failed to provide the initial guess. This is related to the assumed form of displacement, equations (1) and (2), which can only describe small rotations. In practice, this limitation can again be overcome using intermediate images. The present analysis intentionally did not employ the intermediate images. Figure 8(b) shows a comparison of the calculated rotation value versus the prescribed rotation. For both Figs. 8(a) and (b), the errors are on the order of 10^{-13} (essentially the numerical precision), indicating the correct implementation of the relevant parts of the overall DIC algorithm.

Case Study: Heterogeneous Deformation

Ncorr was validated using the entire set of simulated images from SEM 2D-DIC challenge (see Supplementary Material section). Sample Set 14 is a representative example that consists of a reference image as well as three deformed images with a known sinusoidal deformation with increasing strain gradient and imposed noise. Figure 9 shows the horizontal displacement (u) of Sample 1 and 3 with the smallest and largest strain gradient respectively. The analysis was performed using a subset radius of 20 pixels.

Sample 14 highlights many of the issues encountered in applications of DIC. A typical size of the speckle pattern, as assessed from the reference image power spectrum, is approximately 8 pixels (see Supplementary Material). This poses a limitation on the subset size, since the latter must contain enough distinct features to yield a good correlation. Furthermore, smaller subset radii tend to be affected by the noise present in each simulated image. On the other hand, large subset or strain window sizes tend to smoothen the field to the extent that the localized deformation may not be correctly captured. Table 1 shows a summary of the errors obtained using various subset and strain window sizes for each of the three samples in Set 14. To further illustrate this analysis, Fig. 10 shows the profiles of the horizontal displacement $u(x)$ of Sample 1 and Sample 3 averaged over y for three different subset radii $r = 10,\ 20$ and 30 pixels together with the prescribed displacement. The error bars show the standard deviation along the y-direction. The smallest subset radius of 10 pixels contains on average 2.5 distinct features and yields a rather noisy field, as evidenced by the large error bars for both Sample 1 and Sample 3. The effect of the superimposed noise

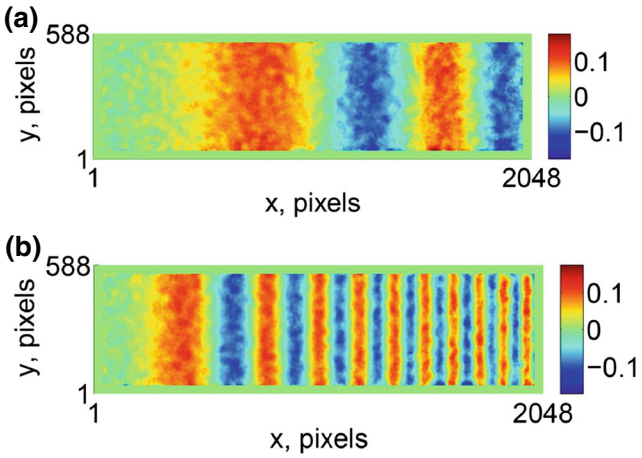

Fig. 9 Horizontal displacement (u) of the heterogeneous deformation of Sample 14 in the SEM 2D DIC challenge. (**a**) Sample 1 and (**b**) Sample 3 with a subset radius r=20 pixels

decreases with increasing subset radius with the subset size of 30 pixels having the smallest deviation from the prescribed displacement. While such large subset sizes are perfectly acceptable for relatively smooth displacement fields in Sample 1, they tend to over-smooth sharp gradients at large x in Sample 3. The effect of subset and strain size on the calculated strains along the horizontal direction is shown in Fig. 11 for Sample 1 and Sample 3. The strain field is significantly noisier than the displacement field with a substantial reduction of the standard deviation with increasing subset and strain window size. However, at the largest subset/strain window ($r = 30$ and $r_E = 20$ pixels) the highly localized deformation of Sample 3 is again not adequately captured. As is so often the case, the DIC analysis involves a trade-off between spatial resolution and resolving small displacements and strains. Therefore, a careful examination of the displacement and strain field for different subset and strain window sizes is necessary in applications to fully assess the quality of the analysis. Ncorr provides the necessary flexibility to vary parameters so that the optimum set for each experiment can be identified.

Case Studies: Experimental Datasets

In this section we demonstrate applications of Ncorr to analysis of experimental datasets from the SEM 2D-DIC challenge and from our own experiments on Compact Tension (CT) polycrystalline nickel superalloy sample under tension. These illustrative examples were chosen because they highlight many of the interesting features of the code (e.g. the ability to analyze complex shaped samples and capability to analyze discontinuous displacement fields), while highlighting the difficulties frequently encountered in the analysis of experimental data. The above images (available for download on the "downloads" section of http://www.ncorr.com) were

Table 1 SEM DIC Sample Set 14 displacement and strain deviation from prescribed for different subset and strain parameters. Here the mean deviation u_0 from is $\langle(u-u_0)^2\rangle^{1/2}$ and that from $\varepsilon_{xx,0}$ is $\langle(\varepsilon_{xx}-\varepsilon_{xx,0})^2\rangle^{1/2}$, where $\langle\ldots\rangle$ denotes average over y direction in the ROI

Sample	r	r_ε	Mean deviation from u_0	Max deviation from u_0	Mean error, u	Mean deviation from $\varepsilon_{xx,0}$	Max deviation from $\varepsilon_{xx,0}$	Mean error, ε_{xx}
1	10	5	4.7×10^{-3}	3.0×10^{-2}	2.8×10^{-2}	7.1×10^{-4}	7.8×10^{-3}	3.4×10^{-3}
1	10	10	4.7×10^{-3}	3.0×10^{-2}	2.8×10^{-2}	4.7×10^{-4}	4.5×10^{-3}	2.0×10^{-3}
1	20	5	2.8×10^{-3}	7.6×10^{-3}	1.1×10^{-2}	2.0×10^{-4}	6.7×10^{-4}	6.3×10^{-4}
1	20	10	2.8×10^{-3}	7.6×10^{-3}	1.1×10^{-2}	1.7×10^{-4}	5.8×10^{-4}	6.3×10^{-4}
1	20	20	2.8×10^{-3}	7.6×10^{-3}	1.1×10^{-2}	1.3×10^{-4}	3.5×10^{-4}	4.1×10^{-4}
1	30	20	2.3×10^{-3}	6.3×10^{-3}	7.1×10^{-3}	8.7×10^{-5}	2.9×10^{-4}	2.3×10^{-4}
2	10	5	5.0×10^{-3}	2.4×10^{-2}	2.8×10^{-2}	7.0×10^{-4}	4.6×10^{-3}	3.4×10^{-3}
2	10	10	5.0×10^{-3}	2.4×10^{-2}	2.8×10^{-2}	4.9×10^{-4}	2.4×10^{-3}	2.0×10^{-3}
2	20	5	3.1×10^{-3}	9.9×10^{-3}	1.1×10^{-2}	2.2×10^{-4}	9.0×10^{-4}	9.5×10^{-4}
2	20	10	3.1×10^{-3}	9.9×10^{-3}	1.1×10^{-2}	2.0×10^{-4}	8.2×10^{-4}	6.4×10^{-4}
2	20	20	3.1×10^{-3}	9.9×10^{-3}	1.1×10^{-2}	1.8×10^{-4}	1.7×10^{-3}	4.1×10^{-4}
2	30	20	4.5×10^{-3}	1.3×10^{-2}	7.6×10^{-3}	2.2×10^{-4}	2.3×10^{-3}	2.4×10^{-4}
3	10	5	5.2×10^{-3}	2.1×10^{-2}	2.9×10^{-2}	6.7×10^{-4}	2.5×10^{-3}	3.4×10^{-3}
3	10	10	5.2×10^{-3}	2.1×10^{-2}	2.9×10^{-2}	4.7×10^{-4}	1.7×10^{-3}	2.0×10^{-3}
3	20	5	5.5×10^{-3}	2.0×10^{-2}	1.1×10^{-2}	3.4×10^{-4}	1.4×10^{-3}	9.5×10^{-4}
3	20	10	5.5×10^{-3}	2.0×10^{-2}	1.1×10^{-2}	3.4×10^{-4}	1.2×10^{-3}	6.3×10^{-4}
3	20	20	5.5×10^{-3}	2.0×10^{-2}	1.1×10^{-2}	4.5×10^{-4}	1.6×10^{-3}	4.0×10^{-4}
3	30	20	1.2×10^{-2}	3.4×10^{-2}	7.3×10^{-3}	7.5×10^{-4}	2.5×10^{-3}	2.4×10^{-4}

analyzed using version 1.2.0 of Ncorr and MATLAB 7.8.0 (R2009a).

Asymmetric Plate Hole of Finite Size Tested in Tension

The SEM DIC challenge contains a dataset for a tension test of a rectangular plate with a circular hole. Here we use this test to demonstrate Ncorr's ability to analyze complex regions of interest (ROI). The advantage of clearly defining a complex boundary is that the subsets closest to the boundary will not necessarily be discarded in the analysis thus giving information up to a subset size close to the boundary. A set of images of a rectangular sample with a hole and unspecified material properties can be found in the SEM 2D DIC challenge website (http://www.sem.org/dic-challenge/). The hole is offset from the plate center and the sample has finite dimensions so that $W \sim 3D$ where W is the sample width and D is the hole diameter. The dataset consists of a reference image and 10 images obtained at increased loading levels. Each image has a size of 400×1040 pixels.

Figure 12 demonstrates Ncorr analysis of this example at two different loading stages (Images 4 and 8). In order to compare the obtained maps with the analytical solution for infinite plate under tension (Panel A), we chose to focus on the principal angle obtained from the Lagrangian strains $\tan(2\theta_p)=E_{xy}/(E_{xx}-E_{yy})$. The principle angle is a sensitive function of all the three strains and does not depend on the material properties, assuming the deformation is in the elastic range of the material. As is apparent from Panel B, the analysis of Image 8 with relatively small subset and strain window sizes captures reasonably well the near-hole field, while the far field differs appreciably from the analytical solution. With larger subset and strain window sizes (Panel C), the near field is no longer resolved and the far-field is somewhat oversmoothed. The strains at an earlier stage of the experiment, shown in Panel E and F are significantly smaller and are poorly resolved by the smaller subset and strain windows. As a consequence, the results for Image 4 are considerably less satisfactory. We note that the all of the results display appreciable asymmetry both in the near and in the far field. In order to further investigate the origin of these discrepancies, we have conducted a number of numerical experiments using a 2D Finite Element Method (FEM) model of an isotropic, linear elastic/perfect plastic solid with similar geometry to the DIC challenge sample. An example of such solution is shown in Panel D of Fig. 12. The numerical analysis suggests that the slight asymmetry in the plate geometry is not sufficient to account for the observed properties of the field. We propose that the more likely source of the asymmetry lies in asymmetric loading conditions.

Fracture Toughness Test of a Polycrystalline Nickel Superalloy

The final example demonstrates a practical application of Ncorr to monitoring the crack-tip evolution during a room

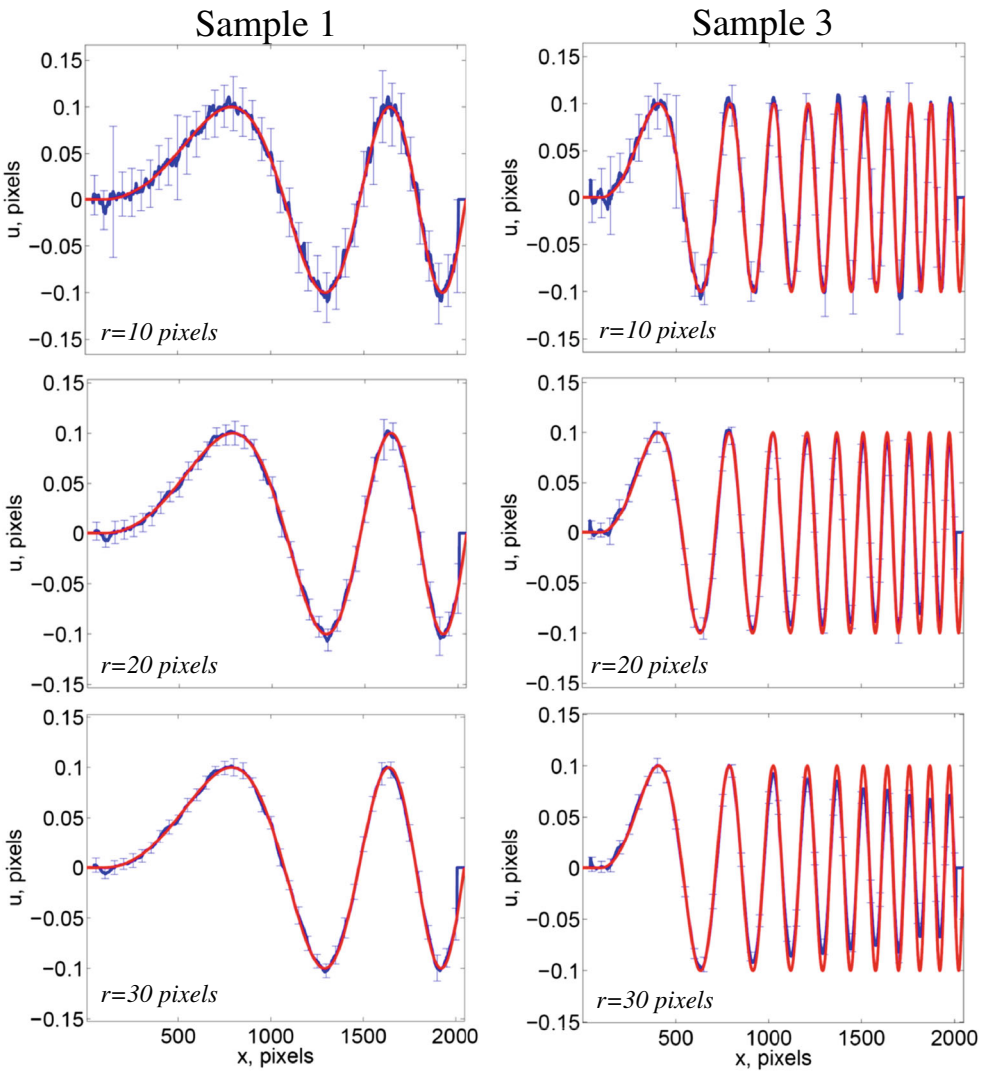

Fig. 10 Horizontal displacement (u) of the heterogeneous deformation of Sample 14 in the SEM 2D DIC challenge. *Left*: Sample 1 and *Right*: Sample 3 for different subset radii r=10–30 pixels. The *red line* is the prescribed displacement. *Blue line* shows the calculated displacement averaged over the y-direction with the standard deviation along y-direction

temperature fracture toughness test of a polycrystalline nickel superalloy.

A Compact Tension (CT) sample of polycrystalline IN100 nickel superalloy [45–47] with an average grain size of 3–5 μm and composition shown on Table 2 was prepared by wire electrical discharge machining (EDM). The CT specimen was polished using 6.5 μm diamond lapping paste. The sample had a width of 30.5 mm, thickness of 2.54 mm, and notch length of 9.5 mm. The sample was cyclically loaded (P_{min}= 170 N; P_{max}~1700 N and R=0.1) until a fatigue crack was initiated. The cyclic experiment continued until the total crack length to sample width ratio was a/W~0.5, where a is the crack length and W is the sample width. For this sample, the fatigue crack was tilted 6° with respect to the horizontal axis.

The sample was subsequently patterned using an in-house patterning apparatus with Montana Gold white acrylic paint. A long distance microscope (LDM) (Questar 100) mounted on a tripod and attached to a camera SBIG 8300 M, which has a high quality full-frame CCD sensor (Kodak KAF-8300) with 16-bit read out and 3326×2504 pixel resolution. The field of view is 5 mm and was aligned so that the crack tip end was close to the center left side of the image. The pattern and microscope optics allowed for both dark-field images and bright field images to be obtained. The dark field images were used in the DIC analysis whereas the bright field images allowed monitoring of the crack tip size and texture to be acquired during a test.

The CT sample was loaded in tension using an MTS servo-hydraulic load frame. At a small load of 90 N the reference image and a rigid body translation (RBT) image were obtained. The RBT image was used to assess the quality of the pattern. The load was increased and bright field and dark field images were obtained at several intervals until fracture occurred at a load of ~6400 N. Bright field images showed that the crack tip length did not change substantially until the sample was close to fracture.

In order to perform the DIC analysis, a mask was formed by uploading a bright field image in Image J software [48] and manually outlining the crack tip length (shown as a red line in

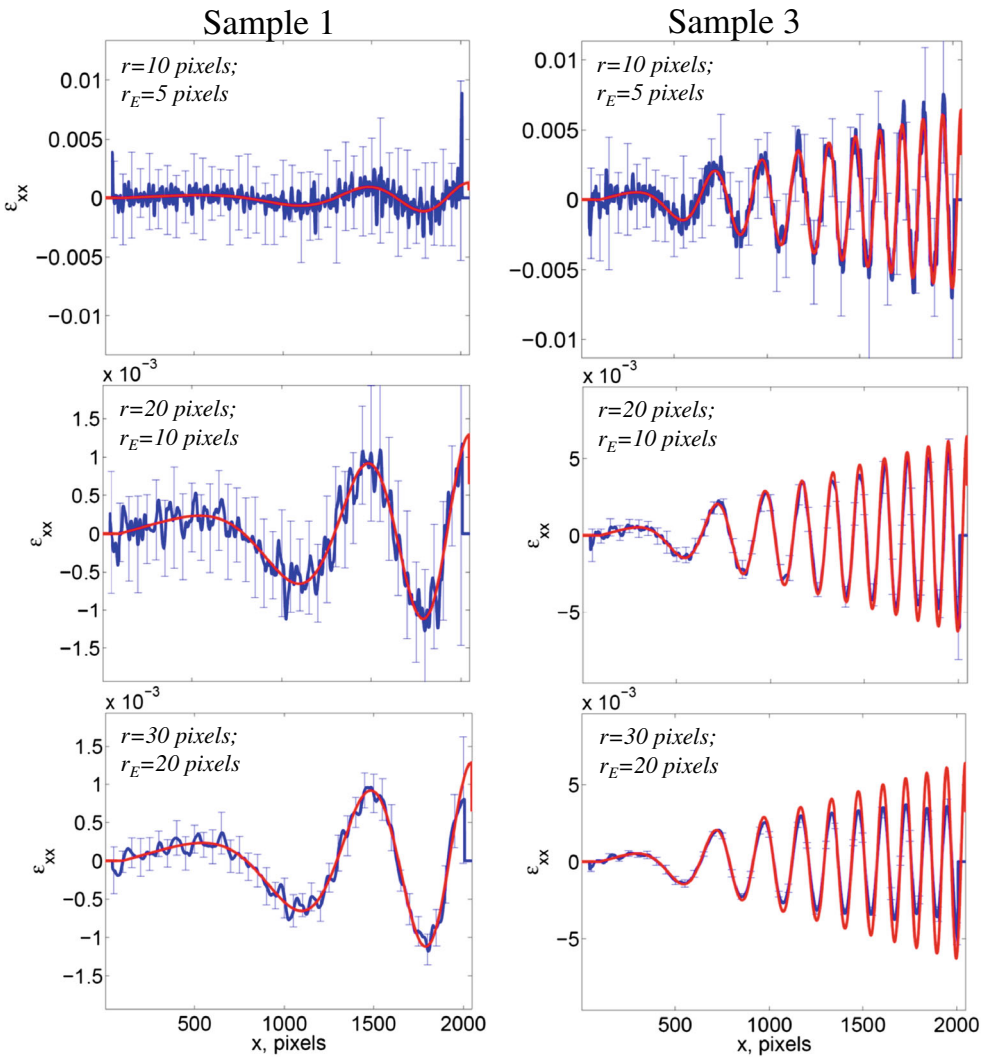

Fig. 11 Normal strain along horizontal direction(ε_{xx}) for Sample 14 in the SEM 2D DIC challenge. *Left*: Sample 1 and *Right*: Sample 3 for different subset radii *(r)* and strain sizes *(r_E)*. The *red line* is the prescribed strain, the *blue line* shows the average strain from Ncorr

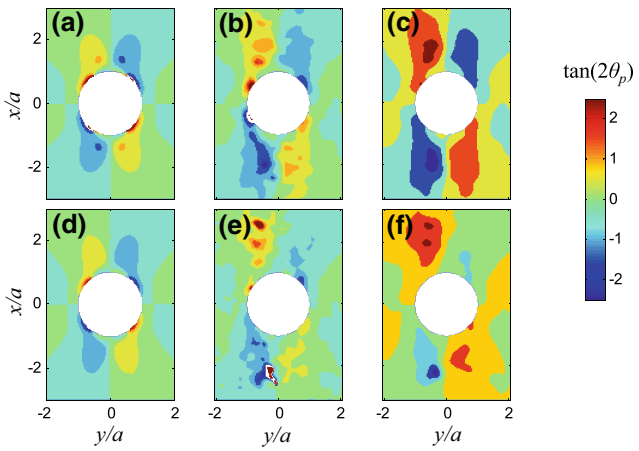

Fig. 12 The near field principal angle parameter for (**a**) analytical solution; (**b**) Image 8 of sample 12 analyzed with subset radius 10 and strain window of 10; (**c**) the same as (**b**), but with subset radius 15 and strain windows of 15; (**d**) FEM model; (**e**),(**f**) Image 4 of sample 12 analyzed with the same parameters as in (**b**) and (**c**). The x, y coordinates are normalized with a, the hole radius

Fig. 13 with the crack tip indicated by the red arrow). This mask was then used as a complex region of interest (ROI) in the DIC analysis. We note that Ncorr has the capability of performing an Eulerian to Lagragian conversion to obtain data within a subset diameter closer to the crack tip region. However, in this example such analysis does not provide substantially different results since the crack-tip length did not change appreciably during loading. While it is still possible to analyze cracked samples by using the "regular" analysis and then filtering out bad points through a correlation coefficient cutoff, any subset that borders a crack will be removed from the data plot. There is a tradeoff between the amount of time required

Table 2 Chemical Composition of IN100 Compact Tension Specimen (% weight)

	Al	B	C	Co	Cr	Mo	Ti	V	Zr	Ni
IN100	4.90	0.02	0.07	18.20	12.10	3.22	4.20	0.70	0.07	56.52

for the analysis and the amount of time needed to trace a complex ROI so that an additional layer of data could be added near the discontinuity.

The effective strain, E_{eff} along with the respective bright field image is shown in Fig. 13(a) and (b) for F~3025 N and in Fig. 13(c) and (d) for F~4100 N. The effective strain is defined as follows: $E_{eff} = \sqrt{2/3(e_{ij}e_{ij})}$ where $e_{ij}=E_{ij}-E_{kk}\delta_{ij}/3$ is the deviatoric component of the Langrangian strain. Both analyses were done with a subset window radius of S_R= 80 pixels (~110 μm), an overlap of 1 pixel, and a strain window radius of 40 pixels. The noise level of the strain field is found to be 10^{-3} as assessed from an RBT image obtained at a small applied load (F~90 N).

As the loading increases, there are two distinct sectors where deformation is localized as seen in the effective strain plot of Fig. 13(c). The largest E_{eff} component occurs at 22° with respect to the horizontal (x-axis). This continues up to $r/a = 0.01$ where the direction of the largest E_{eff} becomes nearly horizontal. The corresponding bright field optical image obtained at F~4100 N shows the two highly localized deformation zones. In this example, the pattern is a limiting factor in the analysis of the strain field. A more refined pattern could be used to resolve the strain near the crack tip and provide more quantitative information on the crack tip evolution.

Summary

This paper documents the core algorithms of Ncorr, a modern open-source DIC software package. Ncorr is freely available to the scientific community and can be easily adapted to suit a

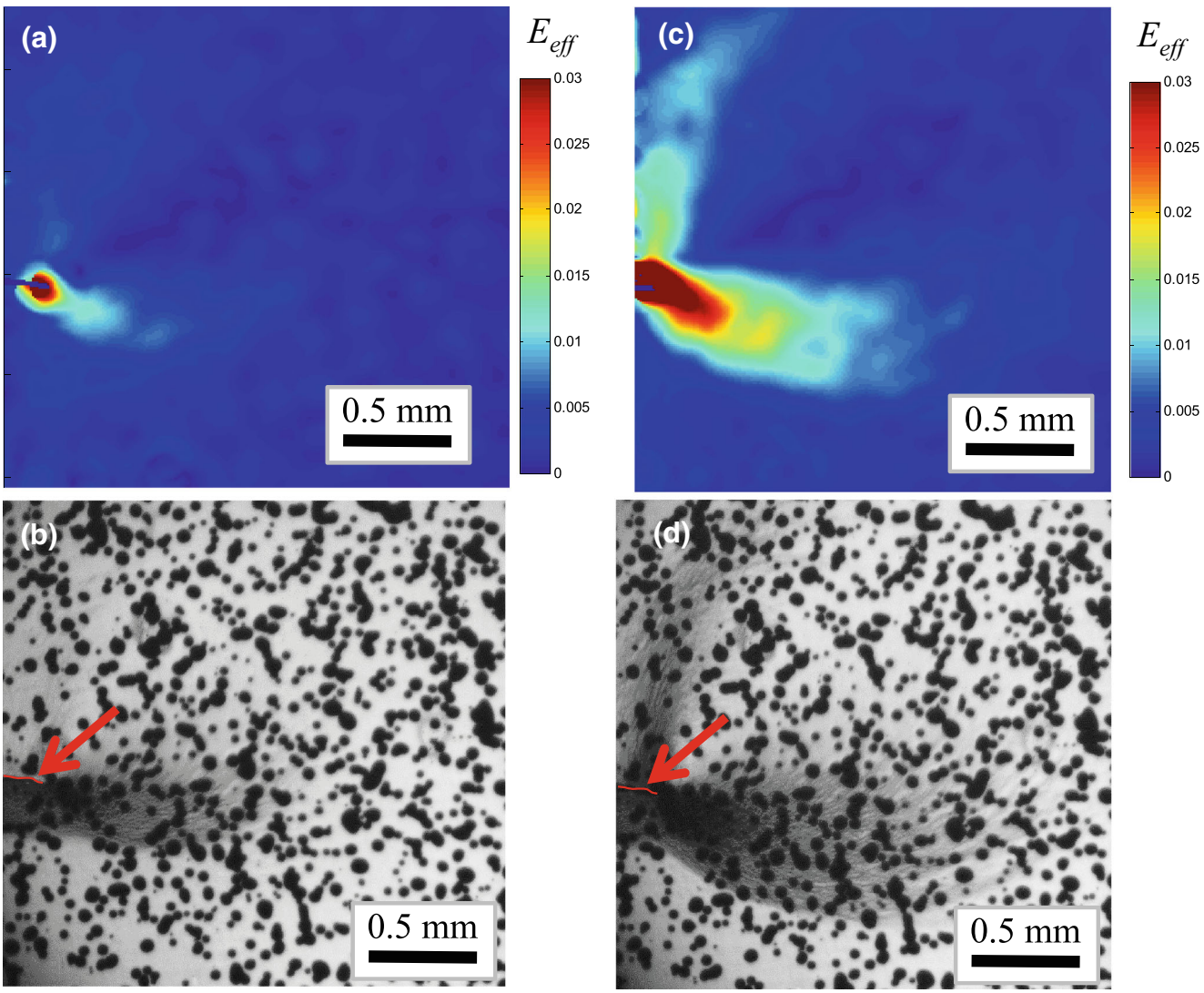

Fig. 13 Polycrystalline nickel superalloy crack tip deformation at (**a**–**b**) F~3025 N and (**c**–**d**) F~4100 N. The effective strain component is shown on the *top row* and the corresponding bright field optical image is shown in the *bottom row*. The contrast and brightness of the bright-field images has been digitally enhanced

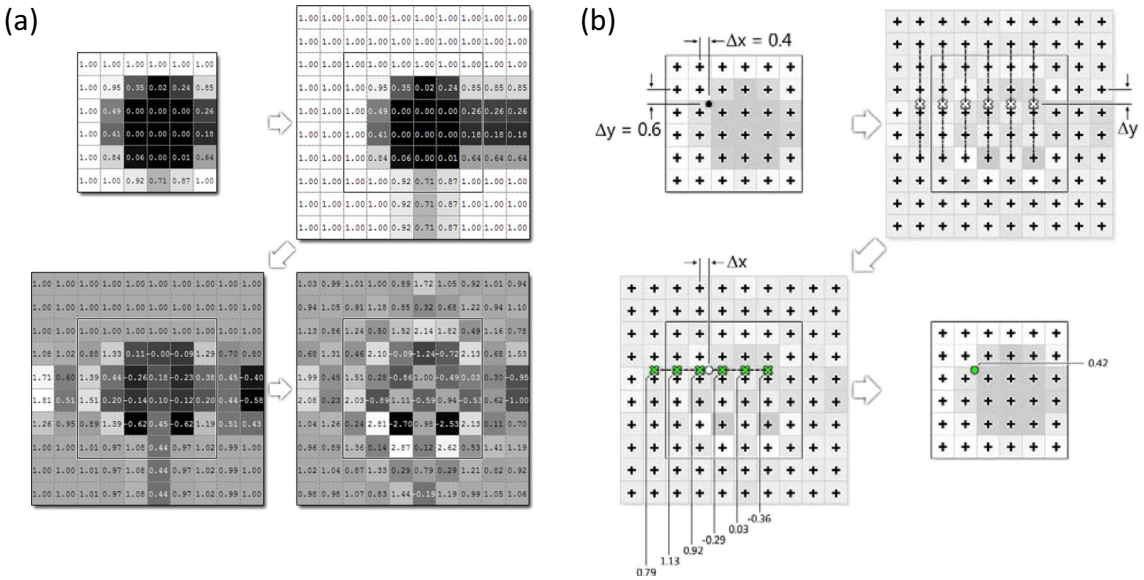

Fig. 14 (**a**) Schematic of the B-spline coefficient calculation. *Top-left*: original grayscale values array. *Top-right*: Copy and pad data; padding parameter here is set to 2. *Bottom-left*: Deconvolution via the DFT for each row and then each column. *Bottom-right*: The associated B-spline coefficients for the top-left image. (**b**) Extension of (**a**) with the same gray scale and B-spline coefficients. *Black crosses* represent integer pixel locations and the *black circle* (*top-left*) is the subpixel point being interpolated

wide range of applications. Several verification tests and sample applications have been described.

Acknowledgments This work has been partially supported by the National Science Foundation (NSF) Graduate Research Fellowship under Grant No. DGE-1148903 and an NSF CAREER Grant No. CMMI-1351705. The fracture toughness test described in this paper was performed at the Mechanical Properties Research Lab at Georgia Tech.

Appendix A1. The Gradient and Hessian Quantities

In order to simplify the calculations, the following assumptions are used

$$\frac{d}{d\boldsymbol{p}}(f_m)\approx 0 \tag{19}$$

$$\frac{d}{d\boldsymbol{p}}\left(\sqrt{\sum\left[f\left(\boldsymbol{\xi}_{\mathrm{ref_c}}+w\left(\Delta\boldsymbol{\xi}_{ref};\mathbf{0}\right)\right)-f_m\right]^2}\right)\approx 0 \tag{20}$$

The gradient for the IC-GN method based on equation (10) is

$$\begin{aligned}\nabla C_{LS}(\mathbf{0}) &= \frac{dC_{LS}(\mathbf{0})}{d\boldsymbol{p}}\\ &\approx \frac{2}{\sqrt{\sum\left[f\left(\boldsymbol{\xi}_{\mathrm{ref_c}}+w\left(\Delta\boldsymbol{\xi}_{ref};\mathbf{0}\right)\right)-f_m\right]^2}}\sum\left[\left[\frac{f\left(\boldsymbol{\xi}_{\mathrm{ref_c}}+w\left(\Delta\boldsymbol{\xi}_{ref};\mathbf{0}\right)\right)-f_m}{\sqrt{\sum\left[f\left(\boldsymbol{\xi}_{\mathrm{ref_c}}+w\left(\Delta\boldsymbol{\xi}_{ref};\mathbf{0}\right)\right)-f_m\right]^2}}\right.\right.\\ &\left.\left.-\frac{g\left(\boldsymbol{\xi}_{\mathrm{ref_c}}+w\left(\Delta\boldsymbol{\xi}_{ref};\boldsymbol{p}_{old}\right)\right)-g_m}{\sqrt{\sum\left[g\left(\boldsymbol{\xi}_{\mathrm{ref_c}}+w\left(\Delta\boldsymbol{\xi}_{ref};\boldsymbol{p}_{old}\right)\right)-g_m\right]^2}}\right]\left[\frac{d}{d\boldsymbol{p}}f\left(\boldsymbol{\xi}_{\mathrm{ref_c}}+w\left(\Delta\boldsymbol{\xi}_{ref};\mathbf{0}\right)\right)\right]\right]\end{aligned} \tag{21}$$

The hessian is

$$\begin{aligned}\nabla\nabla C_{LS}(\mathbf{0}) &= \frac{d^2C_{LS}(\mathbf{0})}{d\boldsymbol{p}^2}\\ &\approx \frac{2}{\sqrt{\sum\left[f\left(\boldsymbol{\xi}_{\mathrm{ref_c}}+w\left(\Delta\boldsymbol{\xi}_{ref};\mathbf{0}\right)\right)-f_m\right]^2}}\left\{\sum\left[\frac{\frac{d}{d\boldsymbol{p}}f\left(\boldsymbol{\xi}_{\mathrm{ref_c}}+w\left(\Delta\boldsymbol{\xi}_{ref};\mathbf{0}\right)\right)}{\sqrt{\sum\left[f\left(\boldsymbol{\xi}_{\mathrm{ref_c}}+w\left(\Delta\boldsymbol{\xi}_{ref};\mathbf{0}\right)\right)-f_m\right]^2}}\right]\left[\frac{d}{d\boldsymbol{p}}f\left(\boldsymbol{\xi}_{\mathrm{ref_c}}+w\left(\Delta\boldsymbol{\xi}_{ref};\mathbf{0}\right)\right)\right]^T\right.\\ &\left.+\sum\left[\frac{f\left(\boldsymbol{\xi}_{\mathrm{ref_c}}+w\left(\Delta\boldsymbol{\xi}_{ref};0\right)\right)-f_m}{\sqrt{\sum\left[f\left(\boldsymbol{\xi}_{\mathrm{ref_c}}+w\left(\Delta\boldsymbol{\xi}_{ref};\mathbf{0}\right)\right)-f_m\right]}}-\frac{g\left(\boldsymbol{\xi}_{\mathrm{ref_c}}+w\left(\Delta\boldsymbol{\xi}_{ref};\boldsymbol{p}_{old}\right)\right)-g_m}{\sqrt{\sum\left[g\left(\boldsymbol{\xi}_{\mathrm{ref_c}}+w\left(\Delta\boldsymbol{\xi}_{ref};\boldsymbol{p}_{old}\right)\right)-g_m\right]^2}}\right]\left[\frac{d^2}{d\boldsymbol{p}}f\left(\boldsymbol{\xi}_{\mathrm{ref_c}}+w\left(\Delta\boldsymbol{\xi}_{ref};\mathbf{0}\right)\right)\right]\right\}\end{aligned} \tag{22}$$

Using the Gauss-Newton assumption

$$\sum\left[\frac{f\left(\boldsymbol{\xi}_{\mathrm{ref_c}}+w\left(\Delta\boldsymbol{\xi}_{\boldsymbol{ref}};\mathbf{0}\right)\right)-f_m}{\sum\left[f\left(\boldsymbol{\xi}_{\mathrm{ref_c}}+w\left(\Delta\boldsymbol{\xi}_{\boldsymbol{ref}};\mathbf{0}\right)\right)-f_m\right]^2}-\frac{g\left(\boldsymbol{\xi}_{\mathrm{ref_c}}+w\left(\Delta\boldsymbol{\xi}_{\boldsymbol{ref}};\boldsymbol{p}_{\boldsymbol{old}}\right)\right)-g_m}{\sqrt{\sum\left[g\left(\boldsymbol{\xi}_{\mathrm{ref_c}}+w\left(\Delta\boldsymbol{\xi}_{\boldsymbol{ref}};\boldsymbol{p}_{\boldsymbol{old}}\right)\right)-g_m\right]^2}}\right]\left[\frac{d^2}{d\boldsymbol{p}^2}f\left(\boldsymbol{\xi}_{\mathrm{ref_c}}+w\left(\Delta\boldsymbol{\xi}_{\boldsymbol{ref}};\mathbf{0}\right)\right)\right]\approx 0 \tag{23}$$

yields the hessian in the final form

$$\nabla\nabla C_{LS}(\mathbf{0})\approx\frac{dC_{LS}(\mathbf{0})}{d\boldsymbol{p}^2}\approx\frac{2}{\sum\left[f\left(\boldsymbol{\xi}_{\mathrm{ref_c}}+w\left(\Delta\boldsymbol{\xi}_{\boldsymbol{ref}};\mathbf{0}\right)\right)-f_m\right]^2}\sum\left[\frac{d}{d\boldsymbol{p}}f\left(\boldsymbol{\xi}_{\mathrm{ref_c}}+\ w\left(\Delta\boldsymbol{\xi}_{\boldsymbol{ref}};\mathbf{0}\right)\right)\right]\left[\frac{d}{d\boldsymbol{p}}f\left(\boldsymbol{\xi}_{\mathrm{ref_c}}+w\left(\Delta\boldsymbol{\xi}_{\boldsymbol{ref}};\mathbf{0}\right)\right)\right]^T \tag{24}$$

Appendix A2. Biquintic B-Spline Interpolation

The quantities $\frac{d}{d\boldsymbol{p}}f\left(\boldsymbol{\xi}_{\mathrm{ref_c}}+w\left(\Delta\boldsymbol{\xi}_{\boldsymbol{ref}};\mathbf{0}\right)\right)$ and $g\left(\boldsymbol{\xi}_{\mathrm{ref_c}}+w\left(\Delta\boldsymbol{\xi}_{\boldsymbol{ref}};\boldsymbol{p}_{\boldsymbol{old}}\right)\right)$ require some form of estimation through interpolation. Using the chain rule on $\frac{d}{d\boldsymbol{p}}f\left(\boldsymbol{\xi}_{\mathrm{ref_c}}+w\left(\Delta\boldsymbol{\xi}_{\boldsymbol{ref}};\mathbf{0}\right)\right)$ and equation (4), we obtain:

$$\frac{d}{d\boldsymbol{p}}f\left(\tilde{x}_{ref_i},\tilde{y}_{ref_j}\right)=\frac{\partial}{\partial\tilde{x}_{ref_i}}f\left(\tilde{x}_{ref_i},\tilde{y}_{ref_j}\right)*\frac{d\tilde{x}_{ref_i}}{d\boldsymbol{p}}+\frac{\partial}{\partial\tilde{y}_{ref_j}}f\left(\tilde{x}_{ref_i},\tilde{y}_{ref_j}\right)*\frac{d\tilde{y}_{ref_j}}{d\boldsymbol{p}} \tag{25}$$

The only two quantities we need to specifically compute for equation (25) are $\frac{\partial}{\partial\tilde{x}_{ref_i}}f\left(\tilde{x}_{ref_i},\tilde{y}_{ref_j}\right)$ and $\frac{\partial}{\partial\tilde{y}_{ref_j}}f\left(\tilde{x}_{ref_i},\tilde{y}_{ref_j}\right)$. These can be computed in various ways (sobel filter, finite difference, etc.), but in Ncorr, biquintic B-spline interpolation is used.

The quantity $g\left(\boldsymbol{\xi}_{\mathrm{ref_c}}+w\left(\Delta\boldsymbol{\xi}_{\boldsymbol{ref}};\boldsymbol{p}_{\boldsymbol{old}}\right)\right)$ also requires interpolation. Once $\frac{\partial}{\partial\tilde{x}_{ref_i}}f\left(\tilde{x}_{ref_i},\tilde{y}_{ref_j}\right)$ and $\frac{\partial}{\partial\tilde{y}_{ref_j}}f\left(\tilde{x}_{ref_i},\tilde{y}_{ref_j}\right)$ are precomputed for the entire reference image, and $g\left(\boldsymbol{\xi}_{\mathrm{ref_c}}+w\left(\Delta\boldsymbol{\xi}_{\boldsymbol{ref}};\boldsymbol{p}_{\boldsymbol{old}}\right)\right)$ is computable, equation (19) and equation (22) can be computed and iterated with equation (10) to find a closer approximation to $\mathbf{p}^*_{\mathrm{rc}}$.

The main idea behind B-spline interpolation is to approximate the image grayscale surface with a linear combination of B-spline basis "splines." These splines are scaled via the B-spline coefficients and then the linear combination of these scaled splines forms an approximation of the surface. Once this approximation is complete, points can be interpolated through 1-D convolutions (since biquintic B-spline interpolation is separable [49]), which reduces to a series of simple dot products. The equation for interpolation for the 1D case is

$$g(x)=\sum_{k\in Z}c(k)\beta^n(x-k) \tag{26}$$

where c(k),β^n(x−k), and g(x) are the B-spline coefficient value at integer k, the B-spline kernel value at x−k, and the interpolated signal value at x, respectively. Here n is the B-spline kernel order, which is set to 5 (the quintic kernel) and Z is the set of integers. Note that the B-spline coefficients are not equivalent to the data samples (unlike in other forms of interpolation—i.e. bicubic keys [50]), and thus must be solved for directly. The equation for the B-spline kernel is

$$\beta^n(x)=\frac{1}{n!}\sum_{k=0}^{n+1}\binom{n+1}{k}(-1)^k\left(x-k+\frac{n+1}{2}\right)_+^n \tag{27}$$

When solved for the quintic case, this equation yields:

$$\beta^5(x)=\begin{cases}\frac{1}{120}x^5+\frac{1}{8}x^4+\frac{3}{4}x^3+\frac{9}{4}x^2+\frac{27}{8}x+\frac{81}{40} & -2\geq x\geq -3\\ -\frac{1}{24}x^5-\frac{3}{8}x^4-\frac{5}{4}x^3-\frac{7}{4}x^2-\frac{5}{8}x^2+\frac{17}{40} & -1\geq x\geq -2\\ \frac{1}{12}x^5+\frac{1}{4}x^4-\frac{1}{2}x^2+\frac{11}{20} & 0\ \ x\geq -1\\ -\frac{1}{12}x^5-\frac{3}{8}x^4+\frac{5}{4}x^3-\frac{7}{4}x^2+\frac{5}{8}x+\frac{17}{40} & 1\ \ x\geq 1\\ \frac{1}{24}x^5-\frac{3}{8}x^4+\frac{5}{4}x^3-\frac{7}{4}x^2+\frac{5}{8}x+\frac{17}{40} & 2\ \ x\geq 1\\ -\frac{1}{120}x^5+\frac{1}{8}x^4-\frac{3}{4}x^3+\frac{9}{4}x^2-\frac{27}{8}x+\frac{81}{40} & 30\ x\geq 2\end{cases} \tag{28}$$

The first step of the interpolation process is to determine the B-spline coefficients. They can be found by using deconvolution. Applying Discrete Fourier Transform (DFT) to equation (26) yields:

$$F\{g\}=F\{c\}*F\{\beta^n\} \tag{29}$$

where $F\{\ldots\}$ is the DFT. The goal is to solve for c, the B-spline coefficients. This can be done by dividing the Fourier coefficients of the B-spline kernel element-wise with the

Fourier coefficients of the signal as shown:

$$F\{c\} = \frac{F\{\beta^{n}\}}{F\{g\}} \tag{30}$$

Taking the inverse DFT of equation (30) will then yield the B-spline coefficients, although caution should be exercised when using this method due to the circular nature of the DFT. To mitigate wrap-around errors, padding should be used.

After obtaining the B-spline coefficients, the image array can be interpolated point-wise by using equation (26). This is carried out by taking a series of dot products with the columns of the B-spline coefficient array and B-spline kernel, and then taking a single dot product across the resulting row of interpolated B-spine coefficient values (note that the order of this operation doesn't matter). The first step of the aforementioned process can be thought of as interpolating the 2D B-spline grid to obtain 1D B-spline coefficient values, and then the second step as interpolating the grayscale value from these 1D B-spline coefficient values.

The steps for obtaining the B-spline coefficients are outlined below:

1. Make a copy of the grayscale array and pad it (any method can be used; Ncorr uses the border values to expand the data as shown in the top right of Fig. 14(a)). Then, sample the B-spline kernel at −2,−1,0,1, 2, and 3. This will form the quintic B-spline vector

$$b_o = \{\, 1/120 \quad 13/60 \quad 11/20 \quad 13/60 \quad 1/120 \quad 0 \,\}^{T} \tag{31}$$

2. Pad the kernel with zeros to the same size as the number of columns (the width) of the image grayscale array. Take the FFT of the padded kernel, and then store it in place.
3. Take the FFT of an image row, then *divide* the Fourier coefficients element-wise of the padded B-spline with the Fourier coefficients from the image row. Afterward, take the inverse FFT of the results and store them in place (in the padded grayscale array). This is done for all the image rows as shown on the bottom right of Fig. 14(a).
4. Repeat steps 2–3, except column-wise, with the array obtained at the end of step 3. The result will be the B-spline coefficients of original image array as shown on the bottom right of Fig. 14(a).

Now that the B-spline coefficients have been obtained, we can interpolate values at sub pixel locations. The steps are outlined below:

1. Pick a subpixel point, $(\widetilde{x}_{cur}, \widetilde{y}_{cur})$, within the image array to interpolate.
2. Calculate Δx and Δy, where:

$$\Delta x = \widetilde{x}_{cur} - x_f \qquad \Delta y = \widetilde{y}_{cur} - y_f \tag{32}$$

   where $x_f = \text{floor}(\widetilde{x}_{cur})$ and $y_f = \text{floor}(\ _{cur})$.
3. Perform the operation in equation (32) to obtain the interpolated grayscale value.

$$g(\widetilde{x}_{cur}, \widetilde{y}_{cur}) = \begin{bmatrix} 1 & \Delta y & \Delta y^2 & \Delta y^3 & \Delta y^4 & \Delta y^5 \end{bmatrix} [QK] [c]_{(x_f-2:x_f+3, y_f-2:y_f+3)} [QK]^T \begin{bmatrix} 1 \\ \Delta x \\ \Delta x^2 \\ \Delta x^3 \\ \Delta x^4 \\ \Delta x^5 \end{bmatrix} \tag{33}$$

where [QK] is the array defined:

$$[QK] = \begin{bmatrix} \frac{1}{120} & \frac{13}{60} & \frac{11}{20} & \frac{13}{60} & \frac{1}{120} & 0 \\ -\frac{1}{24} & -\frac{5}{12} & 0 & \frac{5}{12} & \frac{1}{24} & 0 \\ \frac{1}{12} & \frac{1}{6} & -\frac{1}{2} & \frac{1}{6} & \frac{1}{12} & 0 \\ -\frac{1}{12} & \frac{1}{6} & 0 & -\frac{1}{6} & \frac{1}{12} & 0 \\ \frac{1}{24} & -\frac{1}{6} & \frac{1}{4} & -\frac{1}{6} & \frac{1}{24} & 0 \\ -\frac{1}{120} & \frac{1}{24} & -\frac{1}{12} & \frac{1}{12} & -\frac{1}{24} & \frac{1}{120} \end{bmatrix} \tag{34}$$

and $[c]_{(x_f-2:x_f+3, y_f-2:y_f+3)}$ are the B-spline coefficients as shown:

$$[c]_{(x_f-2:x_f+3, y_f-2:y_f+3)} = \begin{bmatrix} c_{(x_f-2,y_f-2)} & c_{(x_f-1,y_f-2)} & c_{(x_f,y_f-2)} & c_{(x_f+1,y_f-22)} & c_{(x_f+2,y_f-2)} & c_{(x_f+3,y_f-2)} \\ c_{(x_f-2,y_f-1)} & c_{(x_f-1,y_f-1)} & c_{(x_f+1,y_f-1)} & c_{(x_f+1,y_f-1)} & c_{(x_f+2,y_f-1)} & c_{(x_f+3,y_f-1)} \\ c_{(x_f-2,y_f)} & c_{(x_f-1,y_f)} & c_{(x_f,y_f)} & c_{(x_f+1,y_f)} & c_{(x_f+2,y_f)} & c_{(x_f+3,y_f)} \\ c_{(x_f-2,y_f+1)} & c_{(x_f-1,y_f+1)} & c_{(x_f+1,y_f+1)} & c_{(x_f+1,y_f+1)} & c_{(x_f+2,y_f+1)} & c_{(x_f+3,y_f+1)} \\ c_{(x_f-2,y_f+2)} & c_{(x_f-1,y_f+2)} & c_{(x_f,y_f+2)} & c_{(x_f+1,y_f+2)} & c_{(x_f+2,y_f+2)} & c_{(x_f+3,y_f+2)} \\ c_{(x_f-2,y_f+3)} & c_{(x_f-1,y_f+3)} & c_{(x_f,y_f+3)} & c_{(x_f+1,y_f+3)} & c_{(x_f+2,y_f+3)} & c_{(x_f+3,y_f+3)} \end{bmatrix} \tag{35}$$

The position of the required B-spline coefficients within the B-spline array ultimately depends on the amount of padding used. Figure 14(b) gives an example of the location of the coefficients within the B-spline array for a given x_f and y_f and a padding of 2.

The left portion containing the Δy vector and the [QK] matrix is the matrix form of resampling the quintic B-spline kernel with a shift of Δy. Right multiplying this quantity by $[c]_{(x_f=2:x_f+3, y_f-2:y_f+3)}$ yields the interpolated B-spline coefficients which form a row of values, as shown in the top right of Fig. 14(b). When this quantity is right multiplied by [QK] and

the Δx vector, it interpolates the gray-scale value we need from the interpolated row of B-spline coefficients as shown on the bottom left of Fig. 14(b).

Lastly, examining the portion central portion containing:

$$[QK][c]_{\left(x_f-2:x_f+3,y_f-2:y_f+3\right)}[QK]^T \tag{36}$$

this term can be precomputed to increase the speed of the program [37]. This precomputation for biquintic B-spline interpolation requires a very large amount of storage (36 times the size of the padded B-spline coefficient array). But, the space required may be worth the trade off for the speed improvement. The largest computational bottleneck in the DIC analysis is the interpolation step when calculating the components of the hessian, so the reduction in computational time is worth the expensive memory requirement.

At this point, the $g\left(\widetilde{x}_{cur_i},\widetilde{y}_{cur_j}\right)$ quantity is calculable. The last quantities to address are $\frac{\partial}{\partial\widetilde{x}_{ref}}f\left(\widetilde{x}_{ref_i},\widetilde{y}_{ref_j}\right)$ and $\frac{\partial}{\partial\widetilde{y}_{ref}}f\left(\widetilde{x}_{ref_i},\widetilde{y}_{ref_j}\right)$. These quantities can be computed by taking the partial derivatives of an equation of the same form as equation (33) and setting Δx and Δy to zero (because these are integer pixel locations) to obtain

$$\frac{\partial}{\partial\widetilde{x}_{ref}}f\left(\widetilde{x}_{ref_i},\widetilde{y}_{ref_j}\right)=\begin{bmatrix}1&0&0&0&0&0\end{bmatrix}*[QK]*[c]_{\left(x_f-2:x_f+3,y_f-2:y_f+3\right)}*[QK]^T*\begin{bmatrix}0\\1\\0\\0\\0\\0\end{bmatrix} \tag{37}$$

$$\frac{\partial}{\partial\widetilde{y}_{ref}}f\left(\widetilde{x}_{ref_i},\widetilde{y}_{ref_j}\right)=\begin{bmatrix}0&1&0&0&0&0\end{bmatrix}*[QK]*[c]_{\left(x_f-2:x_f+3,y_f-2:y_f+3\right)}*[QK]^T*\begin{bmatrix}1\\0\\0\\0\\0\\0\end{bmatrix} \tag{38}$$

These quantities are precomputed for the entire reference image before beginning the IC-GN method.

References

1. Peters W, Ranson W (1982) Digital imaging techniques in experimental stress analysis. Opt Eng 21(3):213427
2. Chu T, Ranson W, Sutton M (1985) Applications of digital-image-correlation techniques to experimental mechanics. Exp Mech 25(3): 232–244
3. Vendroux G, Knauss W (1998) Submicron deformation field measurements: Part 2. Improved digital image correlation. Exp Mech 38(2):86–92
4. Bruck HA, McNeill SR, Sutton MA, Peters WH III (1989) Digital image correlation using Newton–Raphson method of partial differential correction. Exp Mech 29:261–267
5. Cheng P, Sutton MA, Schreier HW, McNeill SR (2002) Full-field speckle pattern image correlation with B-spline deformation function. Exp Mech 42:344–352
6. Kammers AD, Daly S (2011) Small-scale patterning methods for digital image correlation under scanning electron microscopy. Meas Sci Technol 22:125501
7. Scrivens WA, Luo Y, Sutton MA, Collette SA, Myrick ML, Miney P, Colavita PE, Reynolds AP, Li X (2007) Development of patterns for digital image correlation measurements at reduced length scales. Exp Mech 47(1):63–77. doi:10.1007/s11340-006-5869-y
8. Sutton MA, Li N, Joy DC, Reynolds AP, Li X (2007) Scanning electron microscopy for quantitative small and large deformation measurements. Part I: SEM imaging at magnifications from 200 to 10,000. Exp Mech 47(6):775–787. doi:10.1007/s11340-007-9042-z
9. Wang JW, He Y, Fan F, Liu XH, Xia S, Liu Y, Harris CT, Li H, Huang JY, Mao SX (2013) Two-phase electrochemical lithiation in amorphous silicon. Nano Lett 13(2):709–715
10. Van Puymbroeck N, Michel R, Binet R, Avouac J-P, Taboury J (2000) Measuring earthquakes from optical satellite images. Appl Opt 39(20):3486–3494
11. Rubino V, Lapusta N, Rosakis A, Leprince S, Avouac J (2014) Static laboratory earthquake measurements with the digital image correlation method. Exp Mech 1–18
12. Dickinson AS, Taylor AC, Ozturk H, Browne M (2011) Experimental validation of a finite element model of the proximal femur using digital image correlation and a composite bone model. Engineering, Journal of Biomechanical
13. Zhang D, Eggleton C, Arola D (2002) Evaluating the mechanical behavior of arterial tissue using digital image correlation. Exp Mech 42(4):409–416. doi:10.1007/BF02412146
14. Franck C, Maskarinec SA, Tirrell DA, Ravichandran G (2011) Three-dimensional traction force microscopy: a new tool for quantifying cell-matrix interactions. PLoS One 6(3):e17833
15. Wang H, Lai W, Antoniou A, Bastawros A (2014) Application of digital image correlation for multiscale biomechanics. In: Corey Neu GG (ed) CRC handbook of imaging in biological mechanics. CRC Press, Oxfords, pp 141–151
16. Carroll JD, Abuzaid W, Lambros J, Sehitoglu H (2013) High resolution digital image correlation measurements of strain accumulation in fatigue crack growth. Int J Fatigue 57:140–150
17. Tong W (1997) Detection of plastic deformation patterns in a binary aluminum alloy. Exp Mech 37(4):452–459. doi:10.1007/BF02317313
18. Rehrl C, Kleber S, Antretter T, Pippan R (2011) A methodology to study crystal plasticity inside a compression test sample based on image correlation and EBSD. Mater Charact 62(8):793–800. doi:10.1016/j.matchar.2011.05.009
19. Daly S, Ravichandran G, Bhattacharya K (2007) Stress-induced martensitic phase transformation in thin sheets of Nitinol. Acta Mater 55(10):3593–3600
20. Reedlunn B, Daly S, Hector L, Zavattieri P, Shaw J (2013) Tips and tricks for characterizing shape memory wire part 5: full-field strain measurement by digital image correlation. Exp Tech 37(3):62–78
21. Bastawros A, Bart-Smith H, Evans A (2000) Experimental analysis of deformation mechanisms in a closed-cell aluminum alloy foam. J Mech Phys Solids 48(2):301–322
22. Bart-Smith H, Bastawros A-F, Mumm D, Evans A, Sypeck D, Wadley H (1998) Compressive deformation and yielding mechanisms in cellular Al alloys determined using X-ray tomography and surface strain mapping. Acta Mater 46(10):3583–3592
23. Antoniou A, Onck P, Bastawros AF (2004) Experimental analysis of compressive notch strengthening in closed-cell aluminum alloy foam. Acta Mater 52(8):2377–2386

24. Jerabek M, Major Z, Lang R (2010) Strain determination of polymeric materials using digital image correlation. Polym Test 29(3): 407–416
25. Wang Y, Cuitiño AM (2002) Full-field measurements of heterogeneous deformation patterns on polymeric foams using digital image correlation. Int J Solids Struct 39(13):3777–3796
26. Poissant J, Barthelat F (2008) A novel "subset splitting" procedure for digital image correlation on discontinuous displacement fields. Exp Mech 50:353–364
27. Pan B, Dafang W, Yong X (2012) Incremental calculation for large deformation measurement using reliability-guided digital image correlation. Opt Lasers Eng 50:586–592
28. Pan B, Wang Z, Lu Z (2010) Genuine full-field deformation measurement of an object with complex shape using reliability-guided digital image correlation. Opt Express 18:1011–1023
29. Lu H, Cary PD (2000) Deformation measurements by digital image correlation: implementation of a second-order displacement gradient. Exp Mech 40:393–400
30. Helm JD, McNeill SR, Sutton MA (1996) Improved three-dimensional image correlation for surface displacement measurement. Soc Photo Opt Instrum Eng 35(7):1911–1920
31. Pan B (2009) Reliability-guided digital image correlation for image deformation measurement. Appl Opt 48:1535–1542
32. Eberl C (2010) Digital image correlation and tracking. http://www.mathworks.com/matlabcentral/fileexchange/12413-digital-image-correlation-and-tracking
33. Jones E (2013) Improved digital image correlation (DIC). http://www.mathworks.com/matlabcentral/fileexchange/43073-improved-digital-image-correlation–dic-
34. Pan B, Li K, Tong W (2013) Fast, robust and accurate digital image correlation calculation without redundant computation. Exp Mech 53:1277–1289
35. Pan B, Asundi A, Xie H, Gao J (2009) Digital image correlation using iterative least squares and pointwise least squares for displacement field and strain field measurements. Opt Lasers Eng 47(7):865–874
36. Schreier HW, Braasch JR, Sutton MA (2000) Systematic errors in digital image correlation caused by intensity interpolation. Soc Photo Opt Instrum Eng 39(11):2915–2921
37. Pan B, Li K (2011) A fast digital image correlation method for deformation measurement. Opt Lasers Eng 49:841–847
38. Pan B, Xie H, Wang Z (2010) Equivalence of digital image correlation criteria for pattern matching. Appl Opt 49:5501–5509
39. Baker S, Matthews I (2004) Lucas-kanade 20 years on: a unifying framework. Int J Comput Vis 56(3):221–255
40. Baker S, Matthews I (2004) Lucas-kanade 20 years on: a unifying framework. Int J Comput Vis 56:221–255
41. Pan B (2009) Reliability-guided digital image correlation for image deformation measurement. Appl Opt 408(8):8
42. Eberly D (2000) Least squares fitting of data. Magic Software, Chapel Hill
43. Finley DR (2007) Efficient polygon fill algorithm
44. Nair D, Rajagopal R, Wenzel L (2000) Pattern matching based on a generalized Fourier transform. In: International symposium on optical science and technology. International Society for Optics and Photonics, Bellingham, pp 472–480
45. Milligan W, Orth E, Schirra J, Savage M (2004) Effects of microstructure on the high temperature constitutive behavior of IN100, Superalloys, pp 331–339
46. Jha S, Caton M, Larsen J (2007) A new paradigm of fatigue variability behavior and implications for life prediction. Mater Sci Eng A 468:23–32
47. Barker VM, Johnson SW, Adair BS, Antolovich SD, Staroselsky A (2013) Load and temperature interaction modeling of fatigue crack growth in a Ni-base superalloy. Int J Fatigue 52:95–105
48. Abràmoff MD, Magalhães PJ, Ram SJ (2004) Image processing with ImageJ. Biophoton Int 11(7):36–43
49. Tevenaz P (2000) Interpolation revisited. IEEE Trans Med Imaging 19(7):739–758
50. Keys RG (1981) Cubic convolution interpolation for digital image processing. IEEE Trans Acoust Speech Signal Process 29(6):1153–1160