Antonio Baldi, *Robust algorithms for Digital Image Correlation in the Presence of Displacement Discontinuities* in *Optics and Lasers in Engineering*, 133 (2020), art. n. 106113.

The publisher's version is available at:
https://doi.org/10.1016/j.optlaseng.2020.106113

Robust Algorithms for Digital Image Correlation in the Presence of Displacement Discontinuities

Antonio Baldi
Dipartimento di Ingegneria Meccanica, Chimica e dei Materiali
Università degli Studi di Cagliari
Via Marengo, 2, I-09123 Cagliari, Italy
ORCID:0000-0002-5104-4041
e-mail:`antonio.baldi@dimcm.unica.it`

Abstract

Digital Image Correlation (DIC) is a well-known non-contact experimental technique. Its most common implementation is the subset-based approach, which performs a local least-squares fitting of a simple displacement model — usually an affine transform — to identify the displacement components at the center of the small area under investigation (the subset).

Because of the statistical approach, DIC is usually able to provide reliable results even when theoretical prerequisites are not fully satisfied or in the presence of noise. However, the least-squares algorithm is not robust when the data set contains multiple statistical distributions. Indeed, the algorithm does not discriminate between them and process all the input data; thus, the resulting solution is usually unsatisfactory. A typical example implying the described context is the presence of cracks or shear bands inside the subset: as different sections of the inspected area move in different directions, the algorithm is unable to select a solution and the correlation between the reference and the test image is usually poor.

This work proposes using RANSAC, a well-known robust algorithm, to select the largest domain of the subset. Because a similar problem has to be faced when computing strain components by the polynomial-fitting method, a simple modification of the main algorithm is suggested to handle also this problem.

Keywords: Digital Image Correlation, Cracks, Shear Bands, Strain Estimation, Robust techniques, RANSAC.

1 Introduction

Digital Image Correlation is an optical non-interferometric technique able to estimate the displacement field at a point [1]. It works under the assumption that the intensity of a point of the imaged surface does not change during motion. To allow for recognition of a point, the surface has to be textured and more than one pixel must be considered at the same time, thus, giving rise to a unique pattern. The equation resulting from these assumptions,

$$\frac{\partial I}{\partial x}\dot{u} + \frac{\partial I}{\partial y}\dot{v} + \frac{\partial I}{\partial t} = 0 \tag{1}$$

known as the *optical flow equation*, clearly shows that the displacement identification problem is ill-posed, as it depends on two unknowns.

The standard approach to solving this problem is the Lucas-Kanade formulation [2], which uses an incremental linearized tangent approach: assuming that the intensity field of the test image g is differentiable, the local square residual at point $\mathbf{x}$ can be written as

$$\phi^2(\mathbf{x}) = \{f(\mathbf{x}) - [g(\mathbf{x}) + \mathbf{u}(\mathbf{x}) \cdot \nabla g(\mathbf{x})]\}^2 \tag{2}$$

where we assumed that the intensity of the test image g at point $\mathbf{x} + \mathbf{u}$ is identical to that of reference image f at point $\mathbf{x}$, i.e. $f(\mathbf{x}) = g(\mathbf{x} + \mathbf{u})$. Thus, the cumulative residual over the working domain Ω is

$$\chi^2 = \iint_\Omega \{f(\mathbf{x}) - [g(\mathbf{x}) + \mathbf{u}(\mathbf{x}) \cdot \nabla g(\mathbf{x})]\}^2 \, d\mathbf{x} \tag{3}$$

We assume that the displacement field around the point of interest can be described by a simple polynomial function of local coordinates, thus, $\mathbf{u}(\mathbf{x})$ can be written as:

$$\mathbf{u}(\mathbf{x}) = \sum_\alpha \sum_i p_{\alpha i} \psi_i(\mathbf{x}) \mathbf{i}_\alpha \tag{4}$$

where $\mathbf{i}_\alpha$ are unit vectors along each space dimension α, $\psi_i(\mathbf{x})$ are the shape functions of the displacement field ($\psi = 1, x, y, x^2, \dots$) and $p_{\alpha i}$ are the corresponding weights. Based on previous assumptions, χ^2 can easily be minimized, thus obtaining a linear system $\mathbf{Ap} = \mathbf{b}$:

$$A_{\alpha i \beta j} = \iint_\Omega \psi_i(\mathbf{x}) \psi_j(\mathbf{x}) \, \partial_\alpha g(\mathbf{x}) \, \partial_\beta g(\mathbf{x}) \, d\mathbf{x}$$

$$b_{\alpha i} = \iint_\Omega [f(\mathbf{x}) - g(\mathbf{x})] \, \psi_i(\mathbf{x}) \, \partial_\alpha g(\mathbf{x}) \, d\mathbf{x}$$

where $\partial_\alpha g(\mathbf{x}) = \nabla g(\mathbf{x}) \cdot \mathbf{i}_\alpha$ is the directional derivative of $g(\mathbf{x})$.

Both f and g are highly irregular due to the presence of the texture; thus, the integration is usually implemented as a summation over all pixels in Ω (the local subset of the image).

The described formulation assumes a forward mapping and a Taylor series expansion for the $g(\mathbf{x})$ field. But, this is not the only possible choice and almost all the combinations of *forward* or *backward* mapping, f or g expansion (or

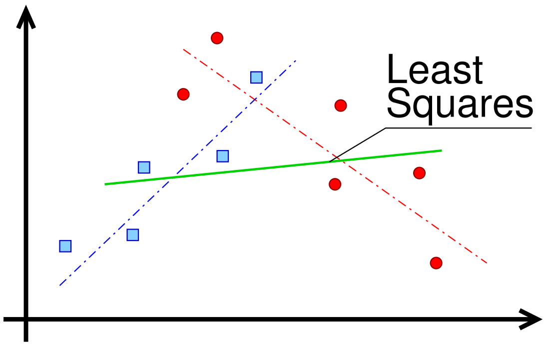

Figure 1: When the data set contains multiple populations (in this case the blue squares and the red circles related to two different linear functions), the least squares algorithm is unable to select one of them and looks for the best solution satisfying both distributions. Consequently, the correlation of the result of the process (the green line) with the experimental data is usually low.

both), have been explored [3–8]. However, disregarding the specific details, all the formulations can be described as the estimation of a local approximation of the displacement field by a nonlinear least-squares approach[1].

The least-squares approach ensures reliable results in the presence of noise. However, the least-squares fitting is not robust against the presence of multiple statistical populations. To give an example, consider the line fitting problem shown in Figure 1. The data set contains more than one population (the blue squares and the red circles, respectively related to two different linear functions); trying to fit all of the data, the algorithm is not able to opt for one of the distributions and the computed interpolating line lyes somewhat in the middle of the two generating functions, i.e. the computed interpolating line does not correlate well with any of the populations.

A similar problem occurs in the Digital Image Correlation context when the active subset is crossed by a crack or a shear band: different parts of the subset follow different displacement laws, thus, (4) is no more a satisfactory displacement model, also considering that summation usually includes only a few terms. As a result, the cumulative residual (3) becomes significantly larger (sometimes by two-three orders of magnitude) and the estimated displacements are no more reliable.

This problem is well known and some solutions have been proposed in the literature. Skipping the ostrich policy (i.e. the deactivation of all subset with a too large χ^2), a few research lines can be identified: Réthoré et al. [14] suggest augmenting the standard shape functions (eq. 4) with crack-related shape

---

[1] Note that to obtain full field information the sketched computation must be repeated several times to completely sample (usually over a regular grid) the region of interest (RoI) of the image. The displacement field can also be described using a global formalism, e.g. by a Finite-Elements-like approach [9–11], B-Splines [12] or NURBS [13].

functions. This approach is shown to be effective in the context of global formulation (i.e. for DIC codes using a Finite-Element-like mesh to describe a global displacement field). The idea has been declined either as an extended FE formulation (xFEM) or integrated DIC (iDIC) or both [15–18]. Still in the context of global DIC formulation, Fagerholt et al. [19] suggest mesh adaption and crack path optimization. Moving to subset-based codes, Helm [20] suggests using quasi-periodic speckle patterns to help the identification of cracks. Finally, Poissant and Barthelat [21] suggest splitting the subset into two parts using a cutting line and a special formulation of the problem. In the last approach, the subset is divided into two regions by the cutting line and the DIC problem is solved for each of them; thus, the minimization process involves the fitting of two independent displacement fields and the cutting line parameters at the same time.

This work proposes using RANSAC [22], a robust regression algorithm, to select the largest population of a subset and perform the analysis on this subsubset of pixels only.

In the next section, the working principle of the algorithm and its implementation in the Digital Image Correlation context will be discussed. Noting that similar problems have to be faced during the post-processing step, we will also discuss strain computation via the plane fitting approach and the use of RANSAC to improve also this point. Finally, the proposed approaches will be validated using numerically generated images and experimental data.

2 RANSAC

To describe the working principle of RANSAC (Random Sample and Consensus), we will still refer to the problem of fitting a line to a data set containing two populations (Figure 2).

Let us assume using a linear model $y = a + bx$ to fit a set of n ($n > 2$) experimental points. As the fitting function involves two parameters, we need only two experimental points to fully identify a solution. Thus, instead of fitting our model, we can randomly select two items in our experimental set and compute a tentative solution $\mathbf{t}_1 = (a_1, b_1)$, where subscript 1 refers to the first function identification.

Because the coefficients of the fitting line are now known, we can easily classify the points, given an error function $\delta(x)$ (e.g. the Euclidean distance from the candidate line), and extract the set of points s inside an admissible band, i.e. the set of points whose absolute error is below the maximum admissible error δ_{max}. Thus, we can rank our tentative solution $\mathbf{t}_1$ by the cardinality of the set s (the number of points in s).

There is nothing special in the first random extraction of points, thus, we can repeat the procedure and randomly select a new couple of points to generate a new tentative solution $\mathbf{t}_2$ and rank it. After a large enough number of iterations n_r, we can assume we found the largest set of points belonging to the population;

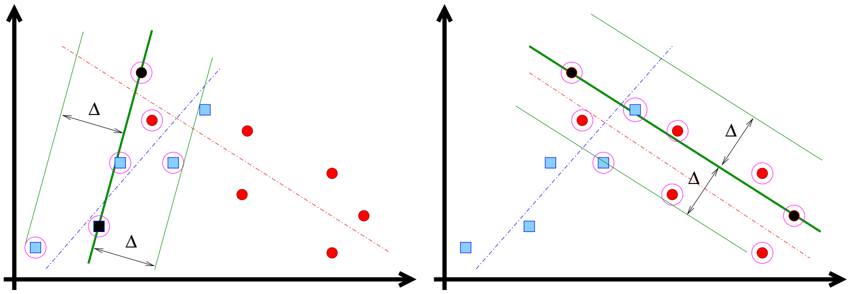

Figure 2: Random Sample and Consensus: (Left) the algorithm randomly peeks the minimum number of points (flagged in black) required for the identification of a candidate solution and computes the number of points within a user tolerance (flagged by magenta circles). After repeating the process several times, the solution with the largest number of candidate points is returned (Right). Please note that a) most, but not all of the selected points belong to the red-circles population; b) the green line is not a best-fit line, which has to be computed a posteriori.

thus, we can use them (and only them) to fit our model.

One may wonder how large n_r is. Rousseeuw and Leroy [23] show that the number of iterations n_r required for giving a probability Γ that a good subsample is selected is

$$n_r = \frac{\log(1-\Gamma)}{\log\left[1-(1-\eta)^p\right]} \tag{5}$$

where η is the fraction of contaminated data and p the number of features in each sample. While p is well defined, η is obviously unknown, so an initial worst guess is required (it cannot be larger than 50%). A fast computation shows that according to (5), less than 12 iterations suffice, even assuming $\Gamma = 99\%$. Our numerical tests show that this estimate is correct from the statistical viewpoint, even though sometimes more iterations are required; in any case, less than 20 iterations are usually more than enough.

Moving the above described sketch to Digital Image Correlation is straightforward: the minimum number of points to be randomly selected for each iteration has to match the number of parameters used in (4) (2 for rigid body translation, 6 for linear polynomials, 12 for parabolic ones, ...) eventually augmented by the number of parameters required for radiosity compensation.

As DIC performs a nonlinear fit to identify a solution, the only general procedure for the identification of a candidate solution (without constraining the displacement field) is using a standard DIC engine working on the list of points selected at the previous step. Finally, equation 2 provides a simple way to check if a given point is near enough to the candidate solution.

The full procedure is described in Algorithm 1: given the list of pixels l of the current subset, we randomly select a minimal-size sub-list w and perform DIC

```
1  procedure RanSAC-displ
       input  : List of points $l$ belonging to current subset
       output: List of active point $a$ of $l$ and its cardinality $c$
2      set $c_{best} \leftarrow 0$, $a_{best} \leftarrow$ empty;
3      for $i \leftarrow 1$ to $N_{iter}$ do
4          build a working subset $w$ by randomly select enough points in $l$
             to uniquely identify DIC shape functions;
5          perform DIC analysis on $w$ to identify a candidate solution;
6          build the list of active points $a$ of $l$ near enough to the candidate
             solution using (2);
7          compute cardinality $c$ of $a$;
8          if $c > c_{best}$ then
9              $c_{best} \leftarrow c$ ;                    // save max cardinality
10             $a_{best} \leftarrow a$ ;                    // ..and list of points
11             if $c > c_{sat}$ then
12                 break;
13             end if
14         end if
15     end for
16     return $a_{best}$ and $c_{best}$;
17 end
```

Algorithm 1: Sketch of RanSAC algorithm for DIC

analysis on it. Knowing the fitted parameters $p_{\alpha i}$ resulting from the analysis of w, we compute the local residue of each pixel of l with (2) and include it in the current set a if $\phi^2(\mathbf{x}) \leq \phi^2_{\text{lim}}$ (where ϕ^2_{lim} is a user-defined limit). If the cardinality of a is larger than the current best value, we update c_{best} and store the list a in a_{best}.

The process is repeated N_{iter} times, selected by (5), and the largest-cardinality list of pixels is returned. Note that we included a simple optimization on line 11 of algorithm 1: if l contains a single population, c will be quite the same as the cardinality of l as early as the first iteration. Thus, we check for this condition (c_{sat} being a large fraction of the cardinality of l, e.g. 90%) and break the iteration.

To complete the algorithm description, we have to define when and how the proposed algorithm should be used. Indeed, even though its use as the "standard" processing algorithm would be harmless, nevertheless such a choice would be highly inefficient from the computational point of view: usually, only a few subsets of the region of interest contain cracks or shear bands; thus, we propose a two steps analysis. Each subset is processed using a standard approach in the former, thus allowing for estimation of the cumulative χ^2 of each subset by (3). RANSAC will only be performed on suspicious subsets, i.e. on subset with $\chi^2 > \chi^2_{\text{lim}}$, where the user-defined χ^2_{lim} is one or two orders of magnitude

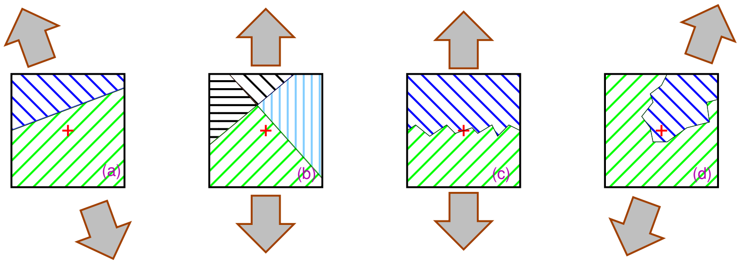

Figure 3: RANSAC can identify the largest population in a subset (in green). Thus, in the presence of a one (a) or more cracks (b), the analysis will be correctly performed. However, the result may be unpredictable if there is no dominant population (c), or it may not corresponds to the motion of the center of subset (d).

larger than the mean χ^2.

Summarizing, we propose using RANSAC to post-process ill-behaved subsets, i.e. subsets showing large χ^2. The presence of cracks or shear bands is not the only reason for such a problem, however, if the bad correlation index is due to the presence of multiple potential solutions, the algorithm is usually able to isolate the largest population (i.e. the largest fraction of pixels following a unique displacement law). Being a statistical approach, we cannot expect it to succeed every time—indeed, results are unpredictable if a crack splits the subset into two equal-size regions (Figure 3-c) or can be simply wrong (Figure 3-d)—but in practice its success ratio is high, even in (the unlikely) case of a subset crossed by multiple cracks (Figure 3-a and b).

3 Computing Strain Components

The result of DIC computation is the set of coefficients $p_{\alpha i}$ describing the displacement field (4) at point $\mathbf{x}$. Thus, providing the shape functions are at least linear, strain components are in principle already available. However, these are usually too noisy to be directly used. To improve strain calculation accuracy, many researchers have proposed performing a smoothing of the displacement field before differentiation: to cite a few, we can recall the Finite-Element-based algorithm proposed by Sutton et al. [12], Meng et al. [24] and Yoneyama [25], a thin-plate spline smoothing algorithm proposed by Wang et al. [26], a univariate natural B-Spline smoothing, a bivariate tensor product spline and the surface averaging approaches proposed by Tong [27], the Local Hermite approach proposed by Li et al. [28] and the regularization approach proposed by Pan et al. [29].

An alternative solution to the strain estimation problem can be obtained either by locally fitting a plane to the experimental data [30, 31] or using Savitzky-Golay filters [32]. These approaches are significantly simpler than the previous ones and allow for control of the smoothing level by selecting the aperture of the computational window; however, they use (directly or implicitly) a least-squares solution. Thus, as in the displacement estimation case, the result may be severely distorted when the sampled data belong to different statistical populations. This situation may occur when the analysis window extends through different regions of a body (e.g. over two sides of a crack) and may appear even though the displacement components are correctly estimated. To overcome such a problem, we can use a simple modification of the previously discussed technique. In the next subsections, we will discuss strain computation by plane fitting (using both the direct least-squares fitting and Savitzky-Golay filters); then, we will introduce the use of RANSAC in this context.

3.1 Strain Computation by Plane Fitting

To compute strain components by plane fitting [30], first, the displacement field is locally fitted (usually with a simple polynomial) within an approximation zone around the point of interest. Then, each differential operator is applied to the interpolant surface.

For relatively small areas, a constant strain approximation can be accepted, thus a linear approximating function, i.e. $d_i = a + b\xi_i + c\eta_i$ suffices, where d equals either u or v and $\xi_i = x_i - x_0$ and $\eta_i = y_i - y_0$ are the coordinate of the sampled points relative to the point of interest (x_0,y_0). Under these assumptions, the linear system resulting from the least-squares fitting of the displacement data is particularly simple:

$$\begin{bmatrix} N & \sum\xi_i & \sum\eta_i \\ \sum\xi_i & \sum\xi_i^2 & \sum\xi_i\eta_i \\ \sum\eta_i & \sum\xi_i\eta_i & \sum\eta_i^2 \end{bmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} \sum d_i \\ \sum d_i\xi_i \\ \sum d_i\eta_i \end{pmatrix} \tag{6}$$

where the summations extend over the neighbors of the point of interest. The resulting b and c are the sought derivative of d respect to x and y (i.e. the strain components at (x_0, y_0)).

The analysis window is expected to be symmetric around the point of interest (it is usually square or circular), but this is not strictly required by the formulation; in the same way, sampling is not required to be performed on a regular grid.

Obviously, the larger the set included in the computation, the smoother the resulting strain field[2]. Thus, the window size may affect strain computation accuracy in the presence of gradients. Frequently-used window sizes range from 11 to 21; an analysis of window size and grid spacing influence on strain accuracy is proposed in [32].

---

[2] It is to be noted that extra smoothing is provided by the way the displacement field is sampled. As the sampling step is usually much smaller than the subset size, neighboring subsets share a large fraction of pixels i.e. the displacement components of neighbor subsets are not independent.

3.2 An Alternative Plane-Fitting Approach: Savitzky-Golay Filters

The above-described approach can be implemented in a formally different way by using Savitzky-Golay two-dimensional filters, which are a class of filters implemented in the time domain. The basic idea is to fit a given polynomial function, e.g.

$$f(\xi, \eta) = c_0 + c_1\xi + c_2\eta + c_3\xi^2 + c_4\xi\eta + \dots \tag{7}$$

to the known displacement components of the working sub-grid i.e., to a regular $(2h+1) \times (2h+1)$ grid of points centered at the point of interest.

The coefficients c_0, c_1, c_2, ... of the best fitting polynomial can easily be found in the least-squares sense by writing (7) for each point of the grid and either solving the resulting rectangular system $\mathbf{Ac} = \mathbf{d}$ by Singular Value Decomposition or using the Moore-Penrose pseudo-inverse. For example, by replacing ξ and η in (7) with the local coordinates of the point located at grid position $(-h, -h)$, we get

$$f(-h, -h) = c_0 + c_1(-hg) + c_2(-hg) + c_3h^2g^2 + \dots = u(-h, -h) \quad \text{or} \quad v(-h, -h)$$

where g is the grid spacing in pixels. Note that both h and g are known integers, thus the terms multiplying the c_0, c_1, ... coefficients are known integer values.

The set of $n = (2h+1) \times (2h+1)$ equations can be organized in matrix form, thus obtaining $\mathbf{Ac} = \mathbf{d}$, where ${}_n\mathbf{A}_m$ is a rectangular matrix (n rows by m columns, where m is the number of polynomial coefficients) and $\mathbf{d}$ is the vector of known displacement components (either u or v) at point. By left-multiplying both sides of equation by $\mathbf{A}^{\mathrm{T}}$ and solving (i.e. by computing the pseudo-inverse $\mathbf{A}^+ = (\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}$), we found a set of m filters (one for each row of ${}_m\mathbf{A}^+_n$), related respectively to the first, second, ..., last polynomial coefficient. Note that the entries of $\mathbf{A}^+$ (i.e. the filter coefficients) do not depend on displacements, thus $\mathbf{A}^+$ can be pre-computed given the filter half-size h and the grid spacing g.

Thus, the previously-described least-squares fitting of a plane reduces to a simple digital convolution when using Savitzky-Golay filters. Some points are worthy of note:

- As we are interested in the first derivative of the polynomial at its origin, we need only the filters related to c_1 and c_2 coefficients. Indeed, as we are computing derivatives in $(0, 0)$, all successive terms are null.

- The coefficients c_0, c_1, ..., c_m can be viewed as coordinates in an m-dimensional Real space with respect to the set of vectors (1, ξ, η, ξ^2, $\xi\eta$, ...). Thus, as the various terms in $f(\xi, \eta)$ constitute a vector space, adding more entries to $f(\xi, \eta)$ does not affect c_1 and c_2 filter masks, which depend only on mask size h and grid spacing g. In other words, fitting the displacement field using a paraboloid does not improve the accuracy of the estimates of first derivatives in the origin.

- The elements of the c_1 and c_2 filters are inversely proportional to g [32]. This is a significant advantage from the computational point of view because filter coefficients can be pre-computed assuming $g = 1$. The correct values can then be estimated by dividing the convolution mask by g.

Summarizing, Savitzky-Golay filters are a very efficient way to estimate the derivatives. However, given the way they are computed, they constitute an alternative (better) way to fit a plane to the displacement field. The price of improved efficiency is the need for a fully populated grid (i.e. they cannot be used near boundaries of the region of interest or whenever one or more points are missing).

```
procedure RanSAC-strain
    input  : List of points $l$ used for polynomial fitting
    output: List of active point $a$ of $l$ and its cardinality $c$
    set $c_{best} \leftarrow 0$, $a_{best} \leftarrow$ empty;
    for $i \leftarrow 1$ to $N_{iter}$ do
        Build a working subset $w$ by randomly select enough points in $l$
          to uniquely identify a fitting function;
        Build the normal system using coordinates of points in $w$;
        Invert the matrix and solve for coefficients of $u$ and $v$ fields
          interpolants;
        Check error of points of $l$ and build the list of active points $a$;
        Compute cardinality $c$ of $a$;
        if $c > c_{best}$ then
            $c_{best} \leftarrow c$ ;                    // save max cardinality
            $a_{best} \leftarrow a$ ;                    // ..and list of points
            if $c > c_{sat}$ then
                break;
            end if
        end if
    end for
    return $a_{best}$ and $c_{best}$;
end
```

Algorithm 2: Sketch of RanSAC algorithm for strain estimation

4 Strain Estimation Using RANSAC

As in the displacement case, the RANSAC algorithm can be used to identify and exclude most of the points not belonging to the dominant population.

The global structure of the algorithm is sketched in algorithm 2 and is similar to the previously described procedure. Note that the sampling grid is going to miss some entries during the process; thus, the Savitzky-Golay filters cannot be used and we have to refer to the standard (slower) least-squares formulation discussed in section 3.1.

To make the described skeleton a working algorithm, we must specify the error function $\epsilon(x, y)$, the error limit $\epsilon_{\max}$ and the activation threshold ϵ_{th}.

Regarding the first point, we have to take into account the numerical-efficiency of the process, because it is going to be executed several times; indeed, the distance of a point P from a plane is well known ($\delta = |c_0 x_p + c_1 y_p + c_2 z_p + c_3|/\sqrt{c_0^2 + c_1^2 + c_2^2}$, where the plane is written in implicit form, i.e. $c_0 x + c_1 y + c_2 z + c_3 = 0$). Although this formulation is invariant with respect to rotations of the coordinate system, it leads to a nonlinear least-squares problem, which has to be solved using an iterative Newton-like algorithm. Thus, in the following, we will assume the error ϵ at point i to be

$$\epsilon(x_i, y_i) = (a + b x_i + c y_i - d_i)^2 \tag{8}$$

because it leads to a linear solution algorithm.

The definition of the limit error $\epsilon_{\max}$ (i.e. of the width of the acceptance band) is a critical point of the RANSAC algorithm. Indeed, the algorithm will be ineffective if $\epsilon_{\max}$ is too large, while a too-tight limit makes it unstable. Assuming that most of the DIC samples are correct (and consequently also the displacement gradients), we propose a two-step approach: in the former, displacement derivatives are computed in all points i of the RoI using a "standard" approach (i.e. either using a plane-fitting algorithm or Savitsky-Golay filters). As a side effect of this process, the normalized cumulative error can be computed:

$$\epsilon_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \left[(a_i^u + b_i^u x_j + c_i^u y_j - u_j)^2 + (a_i^v + b_i^v x_j + c_i^v y_j - v_j)^2 \right] \tag{9}$$

where the suffix u or v refer to the fitted displacement component and both the summation and derivatives in (9) extend to all neighbors of i (i.e. all points originally selected to perform the least-squares fitting/the Savitzky-Golay filtering).

As the normalized error is now known at every point, we can easily compute mean $\langle \epsilon \rangle$ and standard deviation σ_ϵ of the errors; thus, we can assume $\epsilon_{\max} = \langle \epsilon \rangle + \nu \sigma_\epsilon$ with ν being a user-defined scale factor (we adopted $\nu = 1.96$ in our tests, thus assuming that 95% of points are correct).

A similar approach can be used for the computation of the activation threshold of the normalized error ϵ_{th} ($\epsilon > \epsilon_{th}$ activates RANSAC post-processing). In our tests, we used $\epsilon_{th} = \langle \epsilon \rangle + \mu \sigma_\epsilon$ with $\mu = 0.5$. Note that, due to the relatively small value of μ, a large fraction of the active subsets will be validated by the RANSAC algorithm. This is not a problem because the RANSAC/strain iteration involves a solution of a 3×3 linear system only, i.e. the extra-work is very small.

Once $\epsilon_{\max}$ and ϵ_{th} are known, we can check the errors of all points of RoI looking for suspicious points (i.e. points satisfying $\epsilon_i > \epsilon_{th}$) and performing a RANSAC analysis to remove from each subset the points not belonging to the statistical population. The RANSAC-evaluated set of points is finally fitted with a plane to obtain the derivatives.

A few final notes:

- the matrix appearing in (6) may have no inverse (i.e. the selected points are collinear); in this case, it suffices to randomly select a new set;
- it is possible to (marginally) improve performance by LU-decomposing the 3×3 matrix related to plane identification and then back-substitute both u and v displacement arrays;
- various alternative ranking approaches have been discussed in the literature [23, 33, 34]. We tested the MSAC variant, but no significant improvement was observed.

5 Numerical Validation

The proposed procedure should affect a limited fraction of the active subsets. Thus, the evaluation of the improvement in terms of mean value will always result in a marginal refinement. However, looking at the processed subsets only, the benefit is not negligible because our empirical tests show that the ratio of residues before and after the use of the proposed algorithm is usually of the order of tens or better.

To test the performance of the proposed algorithm we opted for using numerically generated images to ensure complete control of the working parameters. The synthetic-image generator is described elsewhere [11, 35, 36], here we give only a concise summary: to avoid polynomial interpolation, the generator assumes that the speckle field can be described as the superposition of several bell-shaped functions randomly "sprayed" over the image.

$$b(r) = \begin{cases} s\left[1 - (r/\rho)^2\right]^3 & r \leq \rho \\ 0 & \text{elsewhere} \end{cases}$$

where $b(r)$ is the normalized intensity (darkness, depending on background) of the speckle at distance r from its center, ρ is the maximum radius (i.e. $b \geq 0$ when $r \leq \rho$) and s is a scaling factor. The location of the center of the speckle and the value of the controlling parameters are randomly generated (and stored); in particular, the location of the center of the speckle is extracted from a uniform distribution mapped on image size; the speckle radius ρ is generated using a normal distribution (mean value = 2.5 pixel, standard deviation = 2 pixel) and the scale factor s is sampled from a uniform distribution between 0.1 and 1. Note that, differently from a two-dimensional Gaussian, b is identically null when $r \geq \rho$; this avoids introduction of a user-defined threshold to select the list of speckles affecting a given point of the image and ensure that there is no "leakage" in the computation of the mean intensity.

Because of the theoretical description, the intensity field can be sampled at any location without using interpolations; this allows avoidance of any difficulties related to reverse mapping: the speckle field is super-sampled and integrated, i.e. the contribution of each sampling is summed to the intensity of the destination pixels (located using the assumed displacement law).

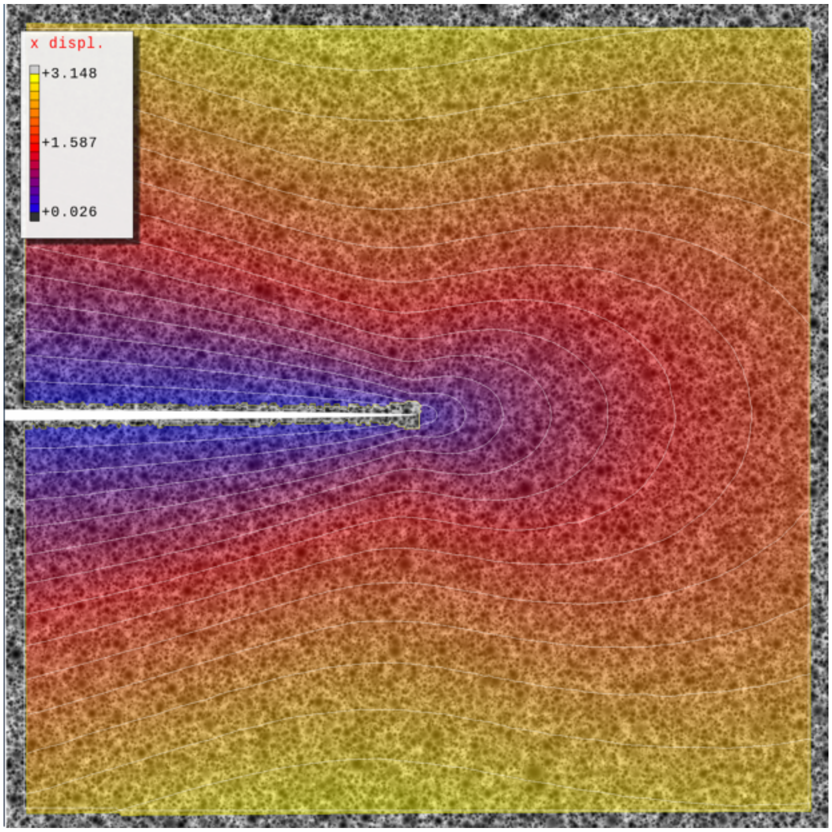

Figure 4: Numerically generated images: u displacement field, mode I crack opening. Fracture opening extends along the x-axis from the left to the center of the picture. Note that the region of interest extends over the full image, i.e. the crack region was not marked in any way.

The complex displacement field $\mathbf{u} = u + iv$ near the crack tip in a semi-infinite body can be expressed [37] as a series of a complex functions $\mathbf{u}(z) = \sum_n [\omega_n \Omega_n(z) + \nu_n \Gamma_n(z)]$, where $z = r\, e^{i\theta}$ and

$$\Omega_n(z) = \frac{(-1)^{\frac{1-n}{2}}}{2\mu\sqrt{2\pi}} r^{\frac{n}{2}} \left\{ \kappa e^{\frac{in\theta}{2}} - \frac{n}{2} e^{\frac{i(4-n)\theta}{2}} + \left[(-1)^n + \frac{n}{2} \right] e^{-\frac{in\theta}{2}} \right\}$$

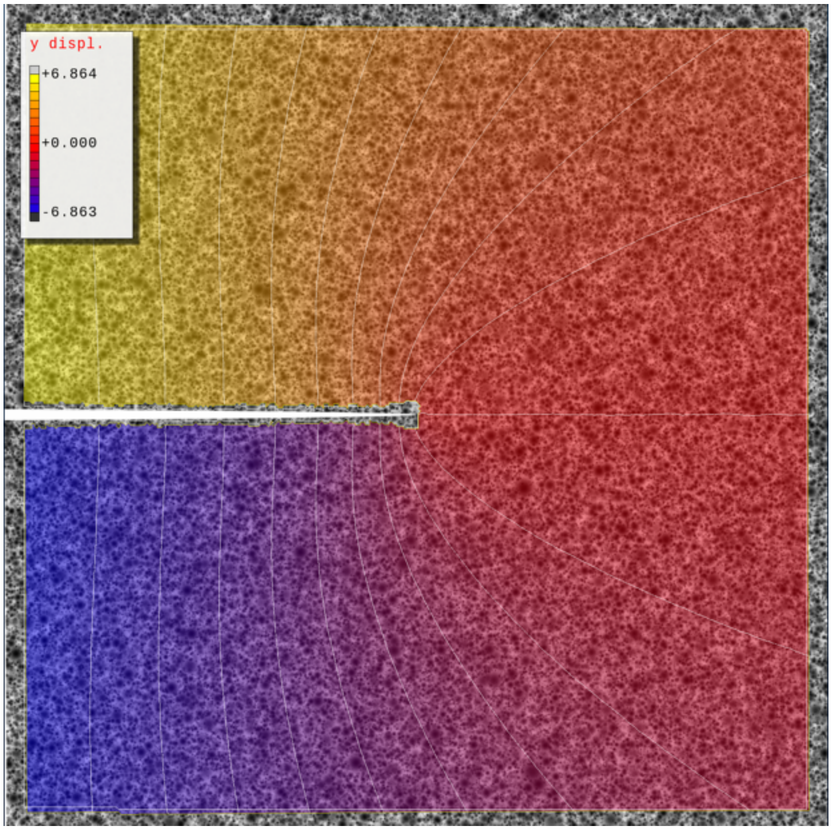

Figure 5: Numerically generated images: v displacement field, mode I crack opening. Note that, despite the discontinuity at crack boundaries, the displacement field is correctly estimated and only a few subsets have been discarded.

for mode I regime, while, for mode II regime

$$\Gamma_n(z) = \frac{i(-1)^{\frac{1-n}{2}}}{2\mu\sqrt{2\pi}} \left\{ \kappa e^{\frac{in\theta}{2}} + \frac{n}{2} e^{\frac{i(4-n)\theta}{2}} + \left[(-1)^n - \frac{n}{2} \right] e^{-\frac{in\theta}{2}} \right\}$$

On truncating the series to the first term and extracting the real and imaginary part, the mode I displacement components related to a reference system with origin at the crack tip and x axis tangent to crack can be expressed as

$$u = \frac{K_I}{2\mu}\sqrt{\frac{r}{2\pi}} \left\{ \cos\left(\tfrac{1}{2}\theta\right) \left[\kappa - 1 + 2\sin^2\left(\tfrac{1}{2}\theta\right) \right] \right\}$$

$$v = \frac{K_I}{2\mu}\sqrt{\frac{r}{2\pi}} \left\{ \sin\left(\tfrac{1}{2}\theta\right) \left[\kappa + 1 - 2\cos^2\left(\tfrac{1}{2}\theta\right) \right] \right\}$$

while mode II displacement components are

$$
\begin{aligned}
u &= \frac{K_{II}}{2\mu}\sqrt{\frac{r}{2\pi}}\left\{\sin\left(\tfrac{1}{2}\theta\right)\left[\ \kappa+1+2\cos^2\left(\tfrac{1}{2}\theta\right)\right]\right\}\\
v &= \frac{K_{II}}{2\mu}\sqrt{\frac{r}{2\pi}}\left\{\cos\left(\tfrac{1}{2}\theta\right)\left[1-\kappa+2\sin^2\left(\tfrac{1}{2}\theta\right)\right]\right\}
\end{aligned}
$$

where μ is the Lamé's second coefficient, ν the Poisson's ratio, r is the distance from the crack tip, θ the angle with respect to the x axis, κ is Kolossov's constant, i.e. $\kappa = 3 - 4\nu$ for plane strain/axialsymmetric problems and $\kappa = (3-\nu)/(1+\nu)$ for plane stress and K_I and K_{II} the mode I and II stress intensity factors.

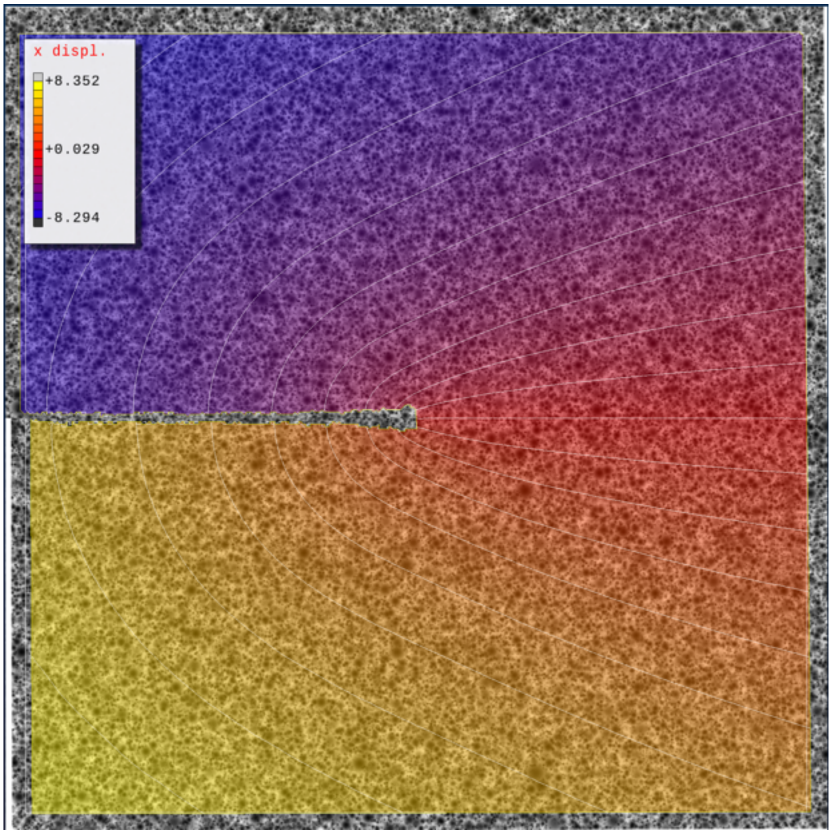

Figure 6: Numerically generated images: u displacement field, mode II crack opening. Note that there is no vertical separation of the two sides of the fracture, but the bottom part slid toward the right with respect to the top area.

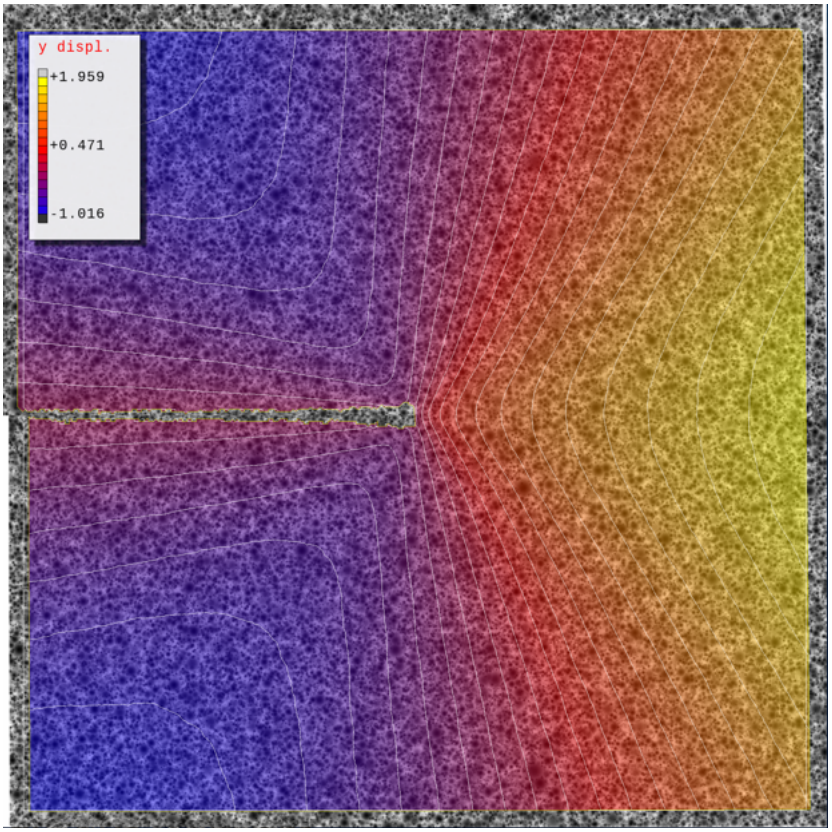

Figure 7: Numerically generated images; v displacement field, mode II crack opening.

Figures 4 and 5 show the displacement fields related to mode I crack opening, while figures 6 and 7 show the same data for mode II. Image size is 1024 pixel×1024 pixel and intensity is quantized using 12 bit. All images have been processed using our in-house developed DIC code using 31×31 pixel^2 subset size with affine shape functions, no low-pass prefiltering, a sampling step of 3 pixel and a 13×13 strain window. The matching criterion was a ZNSSD and the RoI was transversed using a quality-driven algorithm [38–42]. Note that, to emulate real-life conditions, the lower 5 bit of intensity values have been perturbed using uniform distributed noise, while the small sampling step was used to improve results near the edges of the crack.

Figure 8 shows a magnified view of the residues of the DIC analyses related to mode I crack opening when critical subsets (subsets with a large-χ^2) are disabled (left) and when RANSAC is used to pre-process them before disabling (right). The critical threshold for subset disabling was set to $\chi^2 = 5 \times 10^{-4}$,

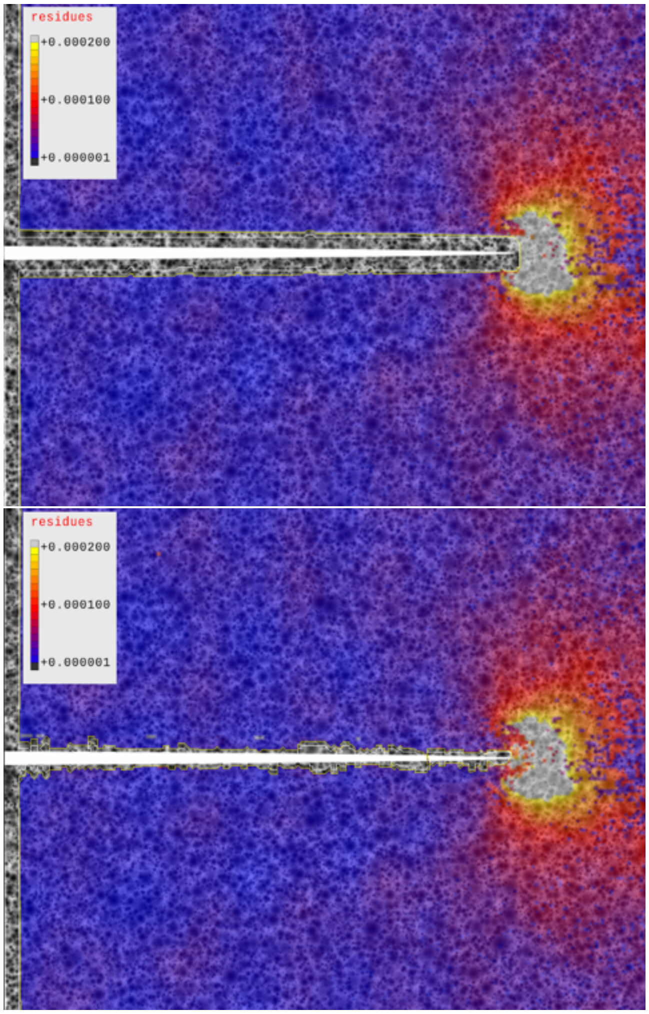

Figure 8: Residues of DIC analysis of mode I problem: cropped area around the crack. Top: results obtained by disabling large-error subsets; bottom: residues using RANSAC and subset disabling. Note that residues are mostly the same in the two images, the only difference being the ability of the RANSAC-enhanced algorithm of processing subsets near the crack. In the same region, the standard algorithm is not able to identify a displacement field correlated enough with the experimental data because of the diverging motion of the opposite sides of the opening, thus code is forced to disable the affected subset to preserve the reliability of results.

but to enhance the visualization, the displayed data range was cropped at $\chi^2 =$

Table 1: Summary of RANSAC execution. Note that not all disabled subsets have been processed by the RANSAC algorithm.

Mode	**Processed** (subsets)	**Recovered** (subsets)	**Disabled** (subsets)	**Strain** (subsets)
I	1335	946	410	704
II	1515	1255	232	512

2×10^{-4}. Looking at the images it is apparent that results are mostly the same[3], apart from the crack edge: when a "standard" approach is used, the residues of subsets near the crack are several orders of magnitude larger than usual value, because subsets contain two sub-regions moving in opposite direction. In such a situation, the only option to preserve accuracy is discarding the subsets, thus, a band of pixels as large as half subset size (or more) is not analyzed at both sides of the crack.

On the contrary, when RANSAC is used, a significant number of subsets are safely recovered by dropping the less significant population. Only subsets where there is no dominant choice are discarded; thus, the analysis extends almost up to the boundaries of the crack. An identical behavior is observed when processing the mode II test: looking at figures 6 and 7, it is apparent that only a few subsets directly over the crack have been discarded, whereas about 1430 more have been preserved using the RANSAC algorithm.

Figure 9 shows the influence of the RANSAC algorithm for strain correction. As discussed in the previous paragraph, the use of the RANSAC algorithm for displacement analysis allows for the preservation of most of the sampling points at crack boundaries. Thus, the strain mask may select points from both sides of the fracture when displacement derivatives are computed at these locations. Figure 9-top shows exactly this behavior for the mode II problem using a pseudo-3d representation. The expected peak at the tip of the fracture (shown in fig. 9-bottom) is actually there, but to allow for visualization of the huge L_{xy} strains at fracture boundaries it is scaled down. Note that, due to the separation, there can be no shear components at crack boundaries. A similar behavior is observed when looking at L_{yy} strain components for the mode I problem (not shown). As in the previous case, the use of the RANSAC algorithm removes the non-existent strains normal to crack boundaries. Table 1 summarizes the number of invocations, of successful executions, of disabled subsets and RANSAC/strain invocations for both mode I and mode II examples.

---

[3] The large out-of-range area is due to asyntotic stress behavior at crack tip

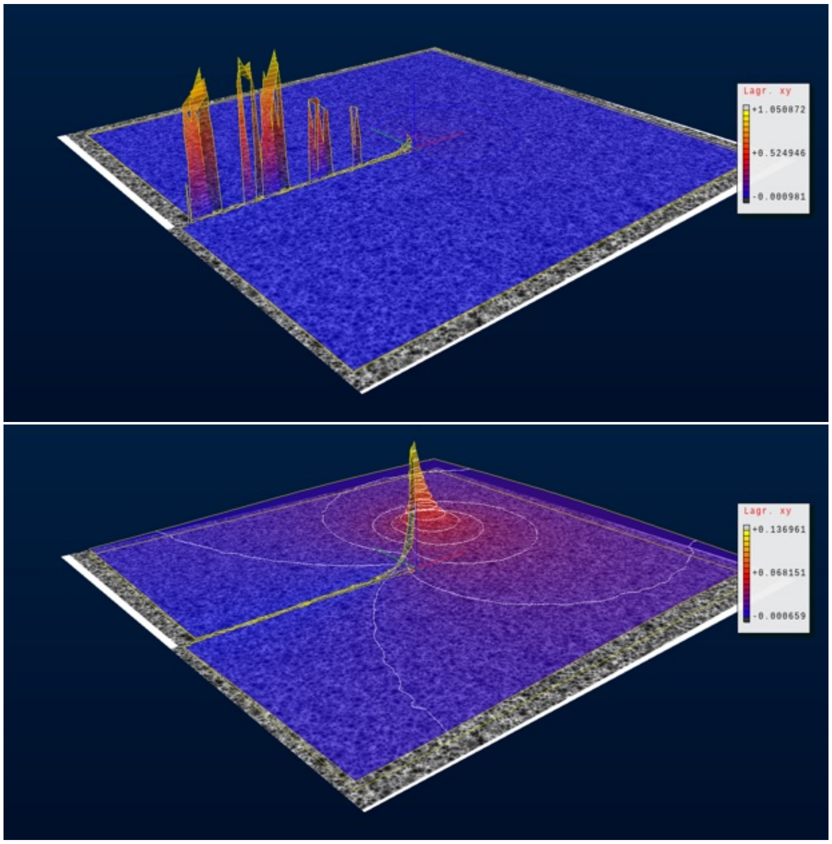

Figure 9: Mode II test, L_{xy} strain (pseudo-3d view). Top: without RANSAC. Bottom: using the RANSAC algorithm for strain estimation. Note the large errors at crack boundaries in the top image.

6 Experimental Test

To test the proposed algorithms in real-life conditions, we performed a tensile test using a speckled background. Figure 10 shows the central area of the specimen and the region of interest (RoI) used during the computation; to enforce localization of the failure, the mid-section of the specimen has been reduced whereas the background is textured to enable computation in this area.

As the RoI extends over the background, we expect to have unreliable subsets (i.e. subsets partially extending over two not-congruent displacement fields) along the left and right boundary of the specimen. A standard processing algorithm should erroneously compute large shear strains during the initial steps—relative displacements between the fixed background and the specimen still being relatively small—coupled with large χ^2. After some steps, the correlation should become so low to make results in the boundary area completely untrust-

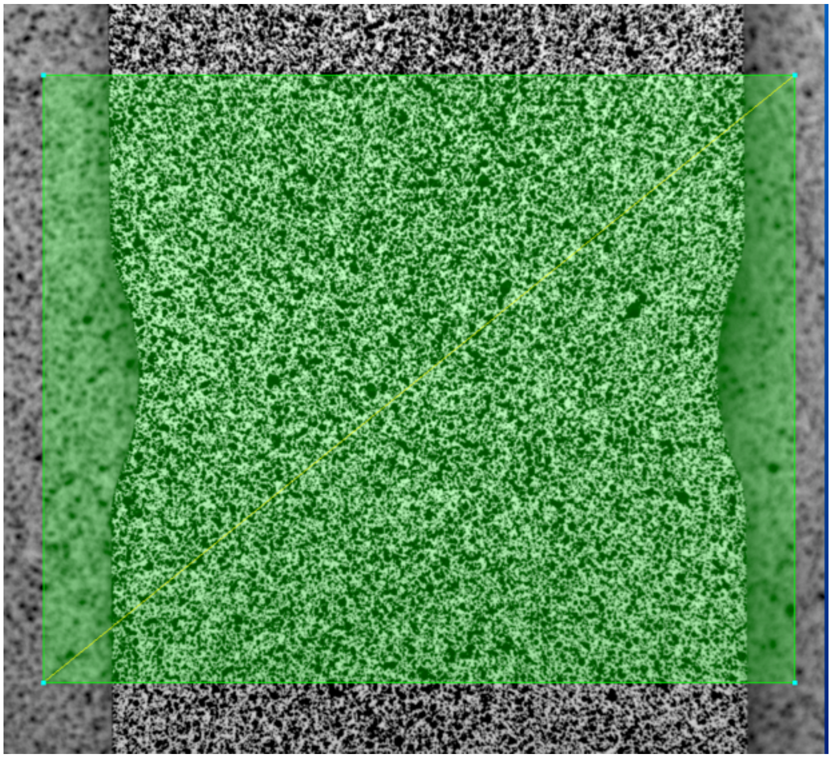

Figure 10: The specimen used for the tensile test. Note that the region of interest extends over the background. The section reduction is used to localize the failure area: due to the local stress intensification, the fracture will be mostly horizontal moving from the center of one notch to the other one.

worthy, thus, a relatively large band of disabled subsets should appear.

Figure 11 shows the xy component (in-plane shear) of the Lagrangian strain tensor at the second computational step, i.e. 3 s after the beginning of the test. The Specimen was loaded using an MTS LandMark 370 universal testing machine working in displacement-control mode ($1\,\mathrm{mm\,min^{-1}}$ linear ramp). Images were acquired using an AlliedVision F421-B Pike camera (sensor: Kodak IT CCD KAI4021, size: 2048×2048 pixel^2, max quantization: 14 bit) using 8 bit quantization, a $0.9\times$ telecentric camera lens and an isotropic, spot-free professional illuminator.

Looking at fig. 11-left, the foreseen shear bands are easily identified. They are an artifact of the strain computational algorithm which is using data sampled from both the (fixed) background and the specimen. When RANSAC algorithm is enabled, the bands do not disappear, but are anyway significantly reduced (fig. 11-right): even though the maximum $|L_{xy}|$ is mostly the same, their width is smaller and gradients are much higher (the contour lines are tightly packed),

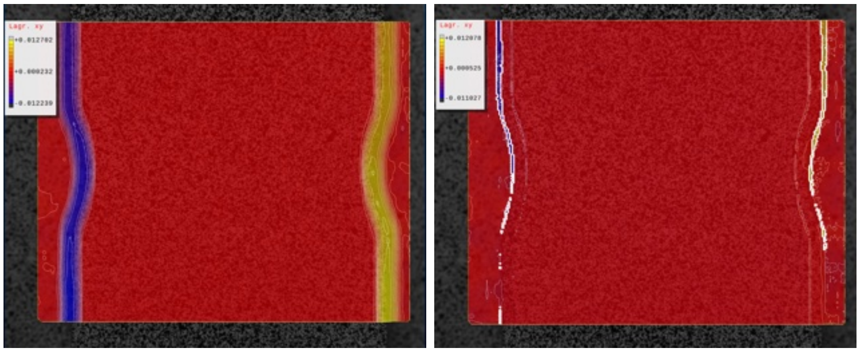

Figure 11: Lagrangian shear strain (L_{xy}) at second step of computation. Left: standard processing; right: using RANSAC. Note the large shear band near the left and right boundary of the specimen when standard processing is used.

thus, ensuring better results near boundaries.

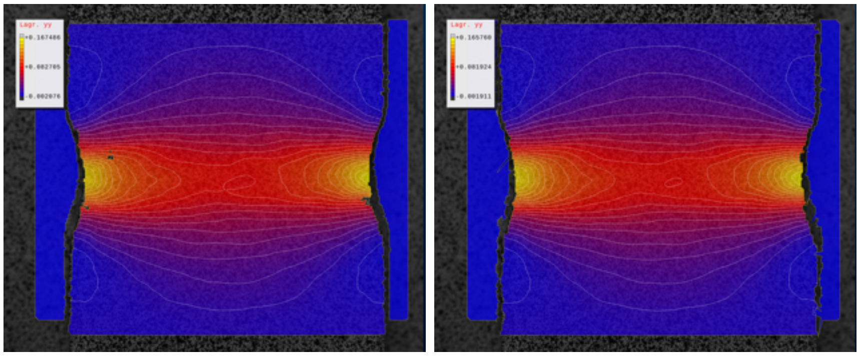

Figure 12: Lagrangian yy strain just before failure. Left: standard processing; right: using RANSAC. The large errors forced the algorithm to disable the subsets.

Figure 12 shows the vertical component of the Lagrangian strain tensor (L_{yy}) a few steps before failure. The comparison of the standard and RANSAC-enabled field points out no significant difference, apart from the RANSAC field being a little bit noisier. This is expected because even though the proposed algorithm allows recovering of a significant part of critical subsets, some of them are anyway disabled. Thus, after several computation steps (the shown images refer to step 99) almost all critical subset have been disabled.

Table 2: Execution time of mode I fracture test with or without RANSAC.

Subsets	**RANSAC** (displ)	**RANSAC** (der.)	**time** [s]	**time** (norm.)	**Notes**
216896	0	0	306.56	1.000	reference
216896	0	4993	312.60	1.020	derivatives only
218134	3435	0	362.16	1.181	displacements only
218134	3435	2895	363.89	1.187	fully enabled

7 Discussion and Conclusions

The previous discussion shows that the introduction of a robust algorithm is beneficial in recovering ambiguous situations, i.e. to identify the dominant displacement field when the subset lies in boundary regions. The improvement is particularly significant in the elastic regime when the correlation index is not deteriorated by various detrimental factors—noise, illumination alteration, paint degradation, large deformations—which makes recovering almost impossible.

Regarding the implementation of the algorithm, we adopted slightly different approaches for RANSAC activation in the displacement- and derivative-estimation case: in the former, we used an user-provided threshold, whereas in the latter we proposed to perform a "standard" analysis, compute the mean and standard deviation of errors and select suspicious subsets basing on the statistical deviation from the mean. Indeed, the statistical approach can be used also in the displacements evaluation case, even when walking the graph using a reliability-guided approach [42]. If a subset analysis failed (i.e. it resulted in a very large χ^2), the subset would be moved to the bottom of the heap and it would not negatively affect the computation of its neighbors. RANSAC would be used at the end of the process, i.e. once the statistical indicators have been estimated. However, if the RANSAC algorithm is processed while walking the graph, it may be able to recover an otherwise useless subset (in the sense that it would not be used to provide hints to its neighbors), thus, potentially improving the solution.

RANSAC is an iterative algorithm. Thus, its computational load may be significant. This is especially true for the displacement processing algorithm because it has to perform a nonlinear fit for each iteration (i.e. a standard DIC evaluation). Table 2 reports the execution time when RANSAC is used either for both evaluation steps (displacements and displacement derivatives) or only one of the two steps. As a reference, we also reported the execution time when RANSAC is not used. Even though the absolute time is not significant[4],

---

[4] The reported value is the so-called *real time*, i.e. the summation of the CPU time over the eight logical cores processing the problem. The machine used was an Intel® Core i7-3770 computer at 3.4 GHz.

the ratio of the execution times with and without RANSAC makes it evident that recovering displacements required 18% extra time. On the contrary, post-processing the displacement derivatives requires a significantly lower computation load (about 2% more). It is interesting to observe that when RANSAC is used also in the displacement step, this value reduces to 0.5%, probably because recovering displacements makes it simpler estimating the derivatives (as is apparent looking at the number of RANSAC invocations for derivatives correction in the second and fourth row).

Looking at timing data reported in the first and third row of table 2, the extra time required by RANSAC for displacement correction is about 55.6 s, i.e. about 39300 DIC evaluations. As the RANSAC-processed subsets are 3435, the mean number of iterations per subset is 11.45. This value is significantly lower than the estimate of (5), which foresees 146 iterations to select a good population ($\Gamma = 0.9$, $\eta = 0.5$, 6 dof), but can be explained by taking into account that some of the subsets selected for RANSAC analysis do not contain multiple populations (thus, the procedure returns in one step) moreover, DIC evaluations performed inside the RANSAC loop involve a minimal number of pixels, thus reducing the computational load with respect to a "standard" DIC analysis.

Summarizing, it makes little or no sense to stay away from the RANSAC algorithm for derivative correction, whereas displacements post-processing may be significant from the computational time viewpoint. In any case, invoking RANSAC on a subset not requiring it is not going to affect results. Indeed, its use is at worst harmless — it will return the previously estimated values at first iteration (i.e. using no significant CPU time) if no outlier is present — but it can significantly enhance performance (reduce error by one or two orders of magnitude) when the points of the sample belong to more than one statistical population.

Finally, concerning the RANSAC algorithm for strain estimation, it is worthy of note that even though in the previous discussion we referred to a linear polynomial, it could be sometimes preferable using a higher-order formulation (parabolic or cubic) depending on the properties of the inspected field. Indeed, in the presence of highly nonlinear displacement fields, a large fraction of the population may be discarded by a linear model, depending on the tolerance used to describe the admissibility band. This would significantly increase the uncertainty of the result, even though its mean value should (in principle) not be affected. The modifications required to move to a higher-order polynomial are minimal: it suffices to update the linear system (6) and the error formulation including higher-order terms; to give an example, assuming a parabolic model, the solution system becomes

$$\begin{bmatrix} N & \sum \xi_i & \sum \eta_i & \sum \xi_i^2 & \sum \xi_i \eta) i & \sum \eta_i^2 \\ \sum \xi_i & \sum \xi_i^2 & \sum \xi_i \eta_i & \sum \xi_i^3 & \sum \xi_i^2 \eta & \sum \xi_i \eta_i^2 \\ \sum \eta_i & \sum \xi_i \eta_i & \sum \eta_i^2 & \sum \xi_i^2 \eta_i & \sum \xi_i \eta_i^2 & \sum \eta_i^3 \\ \sum \xi_i^2 & \sum \xi_i^3 & \sum \xi_i^2 \eta_i & \sum \xi_i^4 & \sum \xi_i^3 \eta_i & \sum \xi_i^2 \eta_i^2 \\ \sum \xi_i \eta_i & \sum \xi_i^2 \eta_i & \sum \xi_i \eta_i^2 & \sum \xi_i^3 \eta_i & \sum \xi_i^2 \eta_i^2 & \sum \xi_i \eta_i^3 \\ \sum \eta_i^2 & \sum \xi_i \eta_i^2 & \sum \eta_i^3 & \sum \xi_i^2 \eta_i^2 & \sum \xi_i \eta_i^3 & \sum \eta_i^4 \end{bmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \\ f \end{pmatrix} = \begin{pmatrix} \sum d_i \\ \sum d_i \xi_i \\ \sum d_i \eta_i \\ \sum d_i \xi_i^2 \\ \sum d_i \xi_i \eta_i \\ \sum d_i \eta_i^2 \end{pmatrix}$$

References

[1] Michael A. Sutton, Jean-José Orteu, and Hubert Schreier. *Image Correlation for Shape, Motion and Deformation Measurements: Basic Concepts,Theory and Applications*. Springer, New York, USA, 2009. ISBN 978-0-387-78746-6. doi: 10.1007/978-0-387-78747-3.

[2] Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of Imaging Understanding Workshop*, volume 130, pages 121–130, 1981.

[3] Juyang Weng, Thomas S. Huang, and Narendra Ahuja. Motion and structure from two perspective views: Algorithms, error analysis, and error estimation. *IEEE transactions on pattern analysis and machine intelligence*, 11(5):451–476, 1989.

[4] C. Quentin Davis and Dennis M. Freeman. Statistics of subpixel registration algorithms based on spatiotemporal gradients or block matching. *Optical Engineering*, 37(4):1290–1298, 1998.

[5] Gary E Christensen and Hans J Johnson. Consistent image registration. *IEEE transactions on medical imaging*, 20(7):568–582, 2001.

[6] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, 2004.

[7] Brian B Avants, Charles L Epstein, Murray Grossman, and James C Gee. Symmetric diffeomorphic image registration with cross-correlation: evaluating automated labeling of elderly and neurodegenerative brain. *Medical Image Analysis*, 12(1):26–41, 2008.

[8] Wei Tong. Formulation of Lucas–Kanade digital image correlation algorithms for non-contact deformation measurements: a review. *Strain*, 49(4): 313–334, 2013.

[9] Y. Sun, J.H.L. Pang, C.K. Wong, and F. Su. Finite element formulation for a digital image correlation method. *Applied Optics*, 44(34):7357–7363, 2005.

[10] G Besnard, François Hild, and Stéphane Roux. “finite-element” displacement fields analysis from digital images: Application to portevin–le châtelier bands. *Experimental Mechanics*, 46:789–803, December 2006. ISSN 0014-4851. doi: 10.1007/s11340-006-9824-8.

[11] Antonio Baldi and Filippo Bertolino. Assessment of h-refinement procedure for global digital image correlation. *Meccanica*, 51(4):979–991, 2016.

[12] M. A. Sutton, J. L. Turner, H. A. Bruck, and T. A. Chae. Full-field representation of discretely sampled surface deformation for displacement and strain analysis. *Experimental Mechanics*, 31(2):168–177, 1991.

[13] Julien Réthoré, Thomas Elguedj, Pierre Simon, and Michel Coret. On the use of NURBS functions for displacement derivatives measurement by digital image correlation. *Experimental Mechanics*, 50(7):1099–1116, 2010.

[14] Julien Réthoré, François Hild, and Stéphane Roux. Extended digital image correlation with crack shape optimization. *International Journal for Numerical Methods in Engineering*, 73:248–272, January 2008. ISSN 00295981. doi: 10.1002/nme.2070.

[15] Stéphane Roux and Franccois Hild. Stress intensity factor measurements from digital image correlation: post-processing and integrated approaches. *International Journal of Fracture*, 140(1):141–157, 2006. doi: 10.1007/s10704-006-6631-2.

[16] Julien Réthoré, Stéphane Roux, and François Hild. An extended and integrated digital image correlation technique applied to the analysis of fractured samples: The equilibrium gap method as a mechanical filter. *European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique*, 18(3-4):285–306, 2009. doi: 10.3166/ejcm.18.285-306.

[17] Stéphane Roux, Julien Réthoré, and François Hild. Digital image correlation and fracture: an advanced technique for estimating stress intensity factors of 2d and 3d cracks. *Journal of Physics D: Applied Physics*, 42(21):214004, 2009.

[18] Florent Mathieu, François Hild, and Stéphane Roux. Identification of a crack propagation law by digital image correlation. *International Journal of Fatigue*, 36(1):146–154, 2012. doi: 10.1016/j.ijfatigue.2011.08.004.

[19] E Fagerholt, T Børvik, and OS Hopperstad. Measuring discontinuous displacement fields in cracked specimens using digital image correlation with mesh adaptation and crack-path optimization. *Optics and Lasers in Engineering*, 51(3):299–310, 2013.

[20] Jeffrey D Helm. Digital image correlation for specimens with multiple growing cracks. *Experimental mechanics*, 48(6):753–762, 2008. doi: 10.1007/s11340-007-9120-2.

[21] J Poissant and F Barthelat. A novel "subset splitting" procedure for digital image correlation on discontinuous displacement fields. *Experimental mechanics*, 50(3):353–364, 2010. doi: 10.1007/s11340-009-9220-2.

[22] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[23] Peter J Rousseeuw and Annick M Leroy. *Robust regression and outlier detection*. John wiley & sons, New York, USA, 1987.

[24] L. B. Meng, G. C. Jin, and X. F. Yao. Application of iteration and finite element smoothing technique for displacement and strain measurement of digital speckle correlation. *Optics and Lasers in Engineering*, 45(1):57–63, 2007.

[25] S Yoneyama. Smoothing measured displacements and computing strains utilising finite element method. *Strain*, 47:258–266, 2011.

[26] Christopher C.B. Wang, Jian-Ming Deng, Gerard A. Ateshian, and Clark T. Hung. An automated approach for direct measurement of two-dimensional strain distributions within articular cartilage under unconfined compression. *Journal of Biomechanical Engineering*, 124(5):557–567, 2002.

[27] Wei Tong. Detection of plastic deformation patterns in a binary aluminum alloy. *Experimental Mechanics*, 37(4):452–459, 1997.

[28] Xin Li, Gang Fang, Jiaqing Zhao, Zhengming Zhang, and Xinxin Wu. Local hermite (lh) method: An accurate and robust smooth technique for high-gradient strain reconstruction in digital image correlation. *Optics and Lasers in Engineering*, 112:26–38, 2019.

[29] Bing Pan, Jieyu Yuan, and Yong Xia. Strain field denoising for digital image correlation using a regularized cost-function. *Optics and Lasers in Engineering*, 65:9–17, 2015.

[30] B. Wattrisse, A. Chrysochoos, J.M. Muracciole, and M. Nemoz-Gaillard. Analysis of strain localization during tensile tests by digital image correlation. *Experimental Mechanics*, 41(1):29–39, 2001.

[31] Bing Pan, Anand Asundi, Huimin Xie, and Jianxin Gao. Digital image correlation using iterative least squares and pointwise least squares for displacement field and strain field measurements. *Optics and Lasers in Engineering*, 47(7-8):865–874, 2009.

[32] Li Bang-Jian, Wang Quan-Bao, and Duan Deng-Ping. Strain measurement errors with digital image correlation due to the Savitzky-Golay filter-based method. *Measurement Science and Technology*, 29(8):085004, 2018.

[33] Philip HS Torr and Andrew Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer vision and image understanding*, 78(1):138–156, 2000.

[34] Peter J Huber. *Robust statistics*. Springer, 2011.

[35] Antonio Baldi and Filippo Bertolino. Experimental analysis of the errors due to polynomial interpolation in digital image correlation. *Strain*, 51(3): 248–263, 2015. doi: 10.1111/str.12137.

[36] Antonio Baldi. Digital image correlation and color cameras. *Experimental Mechanics*, 58(2):315–333, 2018. ISSN 0014–4851. doi: 10.1007/s11340-017-0347-2.

[37] Williams M.L. On the stress distribution at the base of a stationary crack. *ASME Journal Applied Mechanics*, 24:109–114, 1957.

[38] Mitsuo Takeda and Takahide Abe. Phase unwrapping based on maximum cross-amplitude spanning tree algorithm: a comparative study. In *Interferometry VII: Techniques and Analysis*, volume 2544, pages 122–129. International Society for Optics and Photonics, 1995.

[39] Thomas J Flynn. Consistent 2-d phase unwrapping guided by a quality map. In *IGARSS'96. 1996 International Geoscience and Remote Sensing Symposium*, volume 4, pages 2057–2059. IEEE, 1996.

[40] Antonio Baldi, Filippo Bertolino, and Francesco Ginesu. On the performance of some unwrapping algorithms. *Optics and Lasers in Engineering*, 37(4):313–330, 2002.

[41] Xianyu Su and Wenjing Chen. Reliability-guided phase unwrapping algorithm: a review. *Optics and Lasers in Engineering*, 42(3):245–261, 2004.

[42] Bing Pan. Reliability-guided digital image correlation for image deformation measurement. *Applied Optics*, 48(8):1535–1542, 2009.