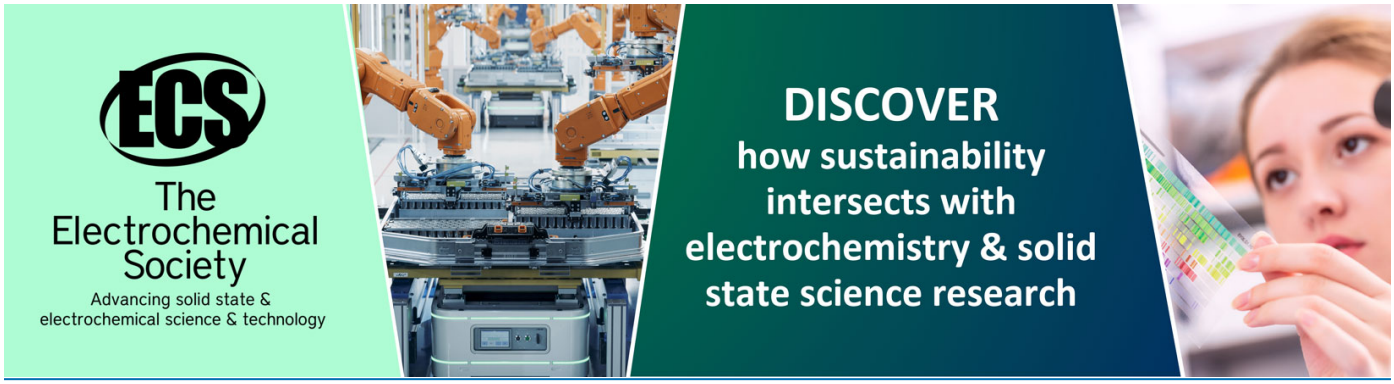
On the Compositional Step of the IC-GN Digital Image Correlation Algorithm



IOP Conf. Series: Materials Science and Engineering 1306 (2024) 012038 doi:10.1088/1757-899X/1306/1/012038

On the Compositional Step of the IC-GN Digital Image Correlation Algorithm

Antonio Baldi[1] **and Pietro Maria Santucci**[1]

[1]Dipartimento di Ingegneria Meccanica, Chimica e Nucleare, Universitá degli Studi di Cagliari, Via Marengo 2, 09123 Cagliari, IT

E-mail: antonio.baldi@unica.it

Abstract. Digital Image Correlation (DIC) is a well-known experimental optical technique for measuring the displacement field based on the assumption that pixel intensity does not change with motion.

DIC can be implemented using various alternative approaches. The most used in the Experimental Mechanics field are the Forward-Additive Gauss-Newton (FA-GN) and the Inverse-Compositional Gauss-Newton (IC-GN) formulations. The former corresponds to the original formulation proposed by Lucas and Kanade, while the latter was proposed by Baker and Matthews twenty years later. Although both formulations give the same results at the first order, their speed, convergency characteristic and noise robustness differ considerably. Nowadays, the IC-GN method is usually preferred because of its lower computational load and the significantly better noise sensitivity. However, the Inverse Compositional approach, as its name states, requires the inversion and composition of the displacement field, thus enforcing the use of invertible displacement fields. This can be a significant limitation because it may introduce an under-matching error in the solution.

This work shows that a simple modification of the viewpoint makes the compositional step simpler, thus giving a DIC formulation as fast as that of the IC-GN, with the same noise-bias characteristic and without requiring an invertible displacement field.

1. Introduction

Digital Image Correlation is a well-known optical, non-interferometric, experimental technique [1]. It works on the assumption that the intensity of a single pixel remains constant during motion. Since the detected intensity includes both the diffuse and reflected components, this apparently simple assumption implies two constraints (uniform illumination and Lambertian surface) [2], which are usually not fully satisfied; however, "standard" experimental setups are usually close enough to the theoretical requirements to make the algorithm work.

The equation resulting from the constant intensity hypothesis, known as the *optical flow equation*,

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

where I is the image intensity and u and v are the displacement components, clearly shows that the displacement identification problem is ill-posed because it involves two unknowns (u and v); without further assumptions, we can only estimate the motion in the direction of the image intensity gradient [3, 4]. Thus, to find a solution, one has either to add a stabilization term [3]

to the differential equation (1) or search for the best *local* solution in the sense of least-squares. The latter is the approach most commonly used in the field of Experimental Mechanics which will be discussed below.

Following the article by Lucas and Kanade [5], several modifications have been proposed [6] to improve solution reliability and computational efficiency. However, they can all be described within the framework of the original algorithm: the *local* mapping between the reference image $f(\mathbf{x})$ and the current image $g(\mathbf{x})$ is assumed to be a simple function of a limited number of parameters, and the solution is found in the sense of least squares. Thus, by considering a sufficiently large number of pixels around the point of interest, it is possible to find the best fit with respect to a given error function. In the following, the Parametric Sum of Squared Differences (PSSD)

$$\chi^2_{PSSD} = \sum_{k=1}^{N} \left[a + bf(x_k, y_k) - g(x_k + u_k, y_k + v_k)\right]^2 \tag{2}$$

is assumed [7], although we will sometimes refer to the Sum of Squared Differences (SSD) criterion

$$\chi^2_{SSD} = \sum_{k=1}^{N} \left[f(x_k, y_k) - g(x_k + u_k, y_k + v_k)\right]^2 \tag{3}$$

to simplify the discussion if the modifications can be easily inferred.

Whatever the criterion (either (2) or (3)), the summation is over all the N pixels of the active area (usually a small subset of the image around the point of interest), while the a and b parameters in (2) take into account the avarage intensity and contrast correction between $f(\mathbf{x})$ and $g(\mathbf{x})$.

A simple polynomial mapping is usually accepted for the u and v displacement components:

$$\begin{aligned} u &= s_0 + s_1\xi + s_2\eta + [s_3\xi^2 + s_4\xi\eta + \dots] \\ v &= t_0 + t_1\xi + t_2\eta + [t_3\xi^2 + t_4\xi\eta + \dots] \end{aligned} \tag{4}$$

where ξ and η are the *local* coordinates of the subset ($\xi = x - x_0$, $\eta = y - y_0$, where x_0, y_0 are the *global* coordinates of the point of interest), s_i and t_i are the set of parameters controlling the mapping (the unknowns of the problem), and the terms in brackets appear only when parabolic shape functions are used.

The solution is identified by a Gauss-Newton approach: the function $g(x + u, y + v)$, which describes the intensity of the current image at point $(x + u, y + v)$, is expanded in Taylor's series truncated to the first order, allowing the calculation of a linear approximation of the local behavior. The χ^2 function (3) thus becomes

$$\chi^2_{SSD} = \sum_{k=1}^{N} \left\{ f_k - \left[g_k + \sum_{i=1}^{m} \left(\frac{\partial g_k}{\partial x}\frac{\partial u}{\partial s_i} ds_i + \frac{\partial g_k}{\partial y}\frac{\partial v}{\partial t_i} dt_i \right) \right] \right\}^2 \tag{5}$$

where we have noted that the displacement component u does not depend on the t_i (v does not depend on s_i) and $du \equiv d\xi \equiv dx$ ($dv \equiv d\eta \equiv dy$).

Computing and setting to zero the derivatives of (5) with respect to ds_i and dt_i (plus a and b in the case of the PSSD criterion) gives a system of $2m$ linear equations ($2(m + 1)$ for PSSD) that can be easily solved using the Cholesky decomposition [8, 9] (the matrix is symmetric and positive definite), thus allowing the updating of the current set of parameters.

Given the new set of parameters, the entire process can be iterated to further refine the solution, until convergence is reached.

The algorithm outlined is known as the Forward-Additive Gauss-Newton (FA-GN) approach, because the displacement components map $f(\mathbf{x})$ to $g(\mathbf{x})$ and the solution is obtained by summing a series of progressively smaller corrections, each obtained by performing one step of the iterative Gauss-Newton algorithm.

Some points are to be noted:

- the least squares fitting approach requires a sufficiently large number of independent data points; so the subset must be large enough (a practical minimum size is around 9×9 pixels) and the surface must have a texture (either natural or artificial);
- we have neglected all second order terms in the Hessian matrix. Although their inclusion may improve convergence near the minimum [10], it does not affect the final result because the second-order derivatives are identically zero when convergence is achieved. In addition, the inclusion of second-order terms makes the algorithm slower and more complex to implement; finally, it can instabilize the algorithm far from the minimum if the model fits the real data poorly or experimental data are contaminated by outlier [9];
- the formulation described assumes a forward mapping and a Taylor series expansion for the $g(\mathbf{x})$ field. In fact, this is not the only possible choice, and all four possible combinations have been studied [6]. Furthermore, *consistent* [11, 12] and *symmetric* [13, 14] formulations have also been proposed; of all these, the Inverse-Compositional Gauss-Newton approach described in the following section seems to be the most widely used because of its noise robustness and numerical efficiency;
- the displacement functions u and v map a point in one of the images to the other one. Thus, the proposed formulation is sometimes described in the literature in terms of *warping functions* (hereafter referred to as $W(\mathbf{x}, \mathbf{p})$ or $\zeta(\mathbf{x}, \mathbf{p})$, where $\mathbf{p}$ is the full set of parameters controlling the function, i.e. $\mathbf{p} = \mathbf{s} \cup \mathbf{t}$).

Regardless of the formulation, DIC requires the comparison of two intensity fields; since the displacements are not restricted to integer values, the intensity of at least one of the images must be interpolated at non-integer locations. In addition, the algorithm requires the calculation of intensity derivatives in both the x and y directions at the same locations. Thus, intensity interpolation and derivative estimation are by far the most computationally intensive steps of DIC. The aim of this work is to identify an efficient and reliable formulation that minimizes these requirements.

2. The Inverse Compositional Approach

The most computationally intensive step of a DIC algorithm is the assembly of the Hessian matrix. Indeed, this step requires the evaluation of the intensity errors and derivatives at each point and for each iteration. Since $u \wedge v \in \mathbb{R}$, at least one of the intensity evaluations of each pair of points requires an interpolation (of either the field $f(\mathbf{x})$ or $g(\mathbf{x})$); however, if the intensity derivatives are computed at fixed locations, it is possible to factor-out this step by pre-evaluating the derivatives before entering the iterative loop: considering that the computation of the derivatives requires approximately the same number of operations of the interpolation, this results in a speed-up by a factor of two.

This objective is achieved by the Inverse Compositional approach [15, 16]. In this algorithm, an incremental parameter update is computed as

$$\Delta\mathbf{u}^{(n)} = \operatorname{argmin} \sum_k \left\{ g\left[W\left(\mathbf{x}_k, \mathbf{p}^{(n)}\right)\right] - f\left[\zeta\left(\mathbf{x}_k, \mathbf{u}_k^{(n)}\right)\right] \right\}^2 \tag{6}$$

where the warping function ζ, related to the reference image, is always an identity, i.e. the shape of the reference subset is reset at the beginning of each iteration. The use of this assumption in

combination with the exchange of roles between the reference and the current image leads to a significant computational improvement. Since the shape functions of the reference image never change, the sampled points are always the same; moreover, due to the exchange of the roles, the computation of intensity derivatives is performed on $f(\mathbf{x})$ (at locations that never change). Thus, the Hessian matrix can be pre-computed before entering the iterative loop[1].

The solution process is as follows:

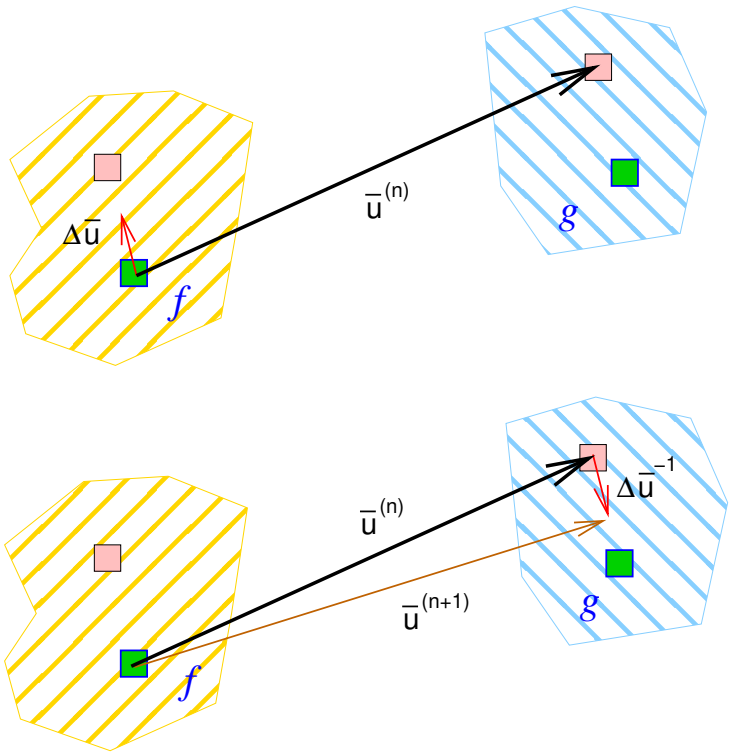


Figure 1: Inverse Compositional mapping. Top: Starting from the reference image f, the green point is coupled with the pink one in the test image g using a forward mapping (should be green). The result of the Gauss-Newton refinement step is the *backward* correction vector $\Delta\mathbf{u}$. Bottom: The backward correction is inverted and composed with the previous vector to produce an improved *forward* estimate (the brown vector).

(i) compute the x and y derivatives of the reference image at the center of the pixels of the subset (i.e. at integer locations);

(ii) compute an initial estimate of the corresponding points in the current image g; thus, the set of parameters $\mathbf{p}$ is related to a *forward* mapping (from f to g);

(iii) using the current forward displacement field, estimate intensity of the pixels of the subset in the current image g (this step requires polynomial interpolation);

(iv) assemble the solution matrix using the pre-computed intensity and derivative of the reference image in combination with the new values of intensities of corresponding points in the current image;

[1] A further significant advantage is that it is possible to use an ad-hoc differential operator working on integer locations.

(v) solve the system. Sincethe roles of f and g are swapped, the solution vector is a *backward* displacement-correction set of parameters;

(vi) since the current set of parameters is related to a forward mapping while the correction set is related to a backward mapping and we are warping both the current and the reference images, it is impossible to use the standard approach and add the correction to $\mathbf{p}$. Instead, the correction must be inverted and composed with the current one, i.e.

$$W\left(\mathbf{x}, \mathbf{p}^{(n+1)}\right) \leftarrow W\left(\mathbf{x}, \mathbf{p}^{(n)}\right) \circ \zeta^{-1}\left(\mathbf{x}, \Delta\mathbf{u}\right)$$

(vii) check for convergence; on failure, go back to step 3, otherwise exit the loop.

Note that $\zeta\left(\mathbf{x}, \Delta\mathbf{u}\right)$ is never updated, so the intensities $f(\mathbf{x}_k)$ and their x and y derivatives $f_x(\mathbf{x}_k)$ and $f_y(\mathbf{x}_k)$, once computed do not need to be updated. The whole process is illustrated in figure 1. In principle, we should move the point in f (figure 1-top); instead, we move the starting point of the "virtual" backward displacement (figure 1-bottom) in such a way that the mapped point of f is always the same. This change of viewpoint takes place in step 6, which is therefore the key point of the algorithm. In fact, the inversion of the incremental displacement and its composition with the current one is a relatively simple operation if a linear mapping is used, because in this case $W(\mathbf{x}, \mathbf{p})$ can be expressed in matrix form as

$$\mathbf{u} = \begin{bmatrix} 1+s_1 & s_2 & s_0 \\ t_1 & 1+t_2 & t_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} \xi \\ \eta \\ 1 \end{pmatrix} \text{ i.e. } \begin{bmatrix} 1+u_x & u_y & u_0 \\ v_x & 1+v_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} \xi \\ \eta \\ 1 \end{pmatrix} \tag{7}$$

Thus, the inversion of the incremental displacement corresponds to a matrix inversion and the composition to a matrix multiplication. However, the former operation constitutes a significant limitation, not so much because of the computational load[2] but because the assumed displacement functions must be invertible. This is a relatively straightforward requirement in the linear case, but only recently has an invertible parabolic function been suggested [18]; to the author's knowledge, there is no formulation for cubic shape functions.

3. From Inverse Compositional to Backward Subtractive

The key point to an efficient implementation of DIC is the pre-computation of the Hessian matrix to avoid evaluating the x and y derivatives of the intensities at all points of the subset for each iteration. As shown in the previous section, this can be achieved by using a backward formulation in combination with the inversion of the incremental displacement[3]. However, this implies the inversion of the incremental displacement matrix and its composition with the current one. The objective of this section is to remove this limitation.

The first step towards this goal is to observe that the displacement function is only used at the beginning of each iteration to identify coupled points in the test and reference images. Thus, we can formally state that the displacement vector starts from g and arrives at f, even though we are actually storing the parameters of the vector from f to g.

The second observation is the fact that the polynomial functions used in (4) form the basis of an m-dimensional real space—$m = 3$ for linear shape functions, 6 for parabolic, 10 for cubic, and so on. Thus, a set of parameters (either s or t) constitutes the coordinate of a point in this space, so that the inverted direction vector can easily be obtained by reversing the sign of all components.

[2] Both matrix multiplication and Cholesky decomposition require $\mathcal{O}(n^3)$ multiplications and a similar number of sums [17], but $n = 3$ in the linear case,

[3] Note that the simple use of a backward formulation is not sufficient, as with this formulation the intensity-evaluation points in f would move, thus requiring a re-computation of $f(\mathbf{x}_k)$ and its derivatives.

Thus, assuming that the roles of reference and current images are swapped (i.e. to adopt a backward formulation), the SSD χ^2 error function becomes

$$\chi^2_{SSD} = \sum_{k=1}^{N} \left[g(x_k, y_k) - f(x_k + u_k, y_k + v_k) \right]^2 \tag{8}$$

where u_k and v_k are the *backward* displacement components ($-u_k$ and $-v_k$ are the forward displacement components). Note that

- unlike the Inverse Compositional approach, we apply the warp function only to the reference image;
- the radiosity compensation has to be applied to g;
- we are working in a purely additive framework.

Expanding $f(x_k + u_k, y_k + v_k)$ in Taylor series, equation (8) becomes

$$\chi^2_{SSD} = \sum_{k=1}^{N} \left\{ g(\mathbf{x}_k) - \left[f(\mathbf{x}_k) + \sum_{i=1}^{m} \left(\frac{\partial f(\mathbf{x}_k)}{\partial x} \frac{\partial u}{\partial s_i} ds_i + \frac{\partial f(\mathbf{x}_k)}{\partial y} \frac{\partial v}{\partial t_i} dt_i \right) \right] \right\}^2$$

Thus, the normal-equation solution matrix $\mathbf{A}$ and the known term vector $\mathbf{b}$ become $\mathbf{A} = \sum_k \mathbf{h}_k \, \mathbf{h}_k^{\mathrm{T}}$ and $\mathbf{b} = \sum_k (f_k - g_k)\mathbf{h}_k$, where

$$\mathbf{h}_k^{\mathrm{T}} = \{-f_x, -f_x\xi, -f_x\eta, \ldots, -f_y, -f_y\xi, -f_y\eta, \ldots\}$$

and the unknown vector is

$$\mathbf{x}^{\mathrm{T}} = \{ds_0, ds_1, \ldots, ds_n, dt_0, dt_1, \ldots, dt_n\}$$

Note that (8) assumes a backward formulation, i.e. u and v are backward displacements and the incremental parameters[4] resulting from solving the system $\mathbf{A}\,\mathbf{x} = \mathbf{b}$ are backward values that should be added to the current backward displacement components to make the point in the reference image f equal to the start value in the current image g. But this would require an update of the target point in the reference frame (i.e. a new calculation of all the f_k of the subset and their derivatives f_x and f_y at non-integer locations). Instead, following the IC-GN approach, we move the starting point in the current image g; but because we are using an additive formulation, and in force of our observation on the vector space, we simply add the sign-reversed incremental vector (figure 2, bottom) instead of computing and composing the inverse.

The resulting algorithm is identical to the Inverse Compositional algorithm, except that step 6 now reads

v ...
vi add $-\Delta\mathbf{p}$ to the current set of parameters, i.e. *subtract* the incremental vector $\Delta\mathbf{p}$ from p (hence the name of the algorithm).
vii ...

Thus, although the algorithm is formulated in terms of backward displacement, we start with (and refine at each step) a set of parameters that relate to forward displacement. The Hessian matrix is computed only at the beginning of the algorithm (before entering the iteration), whereas to perform a new iteration step we only need to interpolate intensity values in g (the current image).

[4] Note that when radiosity compensation is used, the first and second entries in the $\mathbf{x}$ vector are actually total values; indeed, $\mathbf{x}^{\mathrm{T}} = \{a, b, ds_0, ds_1, \ldots, ds_n, dt_0, \ldots, dt_n\}$.

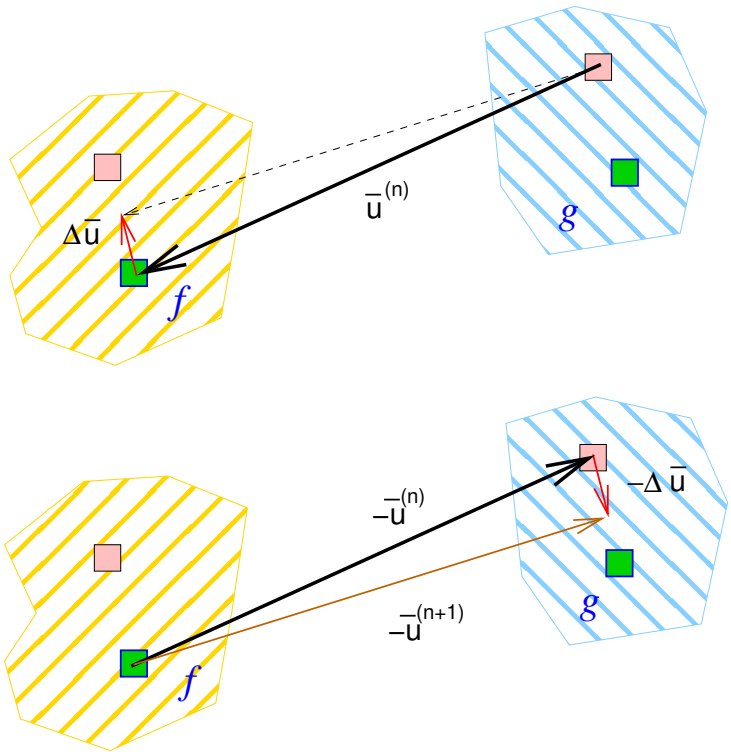


Figure 2: Backward Subtractive algorithm. Above: the algorithm is formulated as a "pure" backward additive algorithm (thus, the subset on image f should move), but (below) we actually store the forward displacement ($-\mathbf{u}$) and add $-\Delta\mathbf{u}$.

4. Equivalence of BS-GN and IC-GN

In [1] the authors show that $\Delta\mathbf{u} = -\Delta\mathbf{p}$ for rigid body motion. In this section we want to show that this is true up to the first order in general and not only for rigid body motion. It is worth noting that near the minimum the second derivatives go to zero, so this condition is sufficient to prove the equivalence.

Assuming that Δu_0, Δu_x, …, Δv_y (Δs_0, Δs_1, …, Δt_2) are the incremental values computed using the backward formulation, the (linear) shape functions associated with a generic step of the Backward Subtractive methods are

$$\mathbf{u} = \begin{bmatrix} 1 + u_x - \Delta u_x & u_y - \Delta u_y & u_0 - \Delta u_0 \\ v_x - \Delta v_x & 1 + v_y - \Delta v_y & v_0 - \Delta v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{pmatrix} \xi \\ \eta \\ 1 \end{pmatrix} \tag{9}$$

where u_0, u_x, …, v_y are the total *forward* values of the displacement components at the start of the step and ξ and η are local coordinates with respect to the center of the subset.

Computing the corresponding matrix for the Inverse Compositional algorithm requires the inversion of the incremental matrix and composition of the result with the current displacement function:

$$\mathbf{u} = \begin{bmatrix} 1 + u_x & u_y & u_0 \\ v_x & 1 + v_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 + \Delta u_x & \Delta u_y & \Delta u_0 \\ \Delta v_x & 1 + \Delta v_y & \Delta v_0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \cdot \begin{pmatrix} \xi \\ \eta \\ 1 \end{pmatrix} \tag{10}$$

The inverse of the incremental matrix can be computed analytically and then multiplied by

the current matrix. The first and second rows of the composed matrix $\mathbf{C} = \mathbf{P}\,\Delta\mathbf{u}^{-1}$, thus result:

$$C(1,1) = \frac{(1+u_x)(1+\Delta v_y)}{d} - \frac{u_y \Delta v_x}{d}$$
$$C(1,2) = \frac{u_y(1+\Delta u_x)}{d} - \frac{(1+u_x)\Delta u_y}{d}$$
$$C(1,3) = u_0 + \frac{u_y(\Delta u_0 \Delta v_x - \Delta u_x \Delta v_0 - \Delta v_0)}{d} + \frac{(1+u_x)(\Delta u_y \Delta v_0 - \Delta u_0 \Delta v_y - \Delta u_0)}{d}$$
$$C(2,1) = \frac{v_x(1+\Delta v_y)}{d} - \frac{(1+v_y)\Delta v_x}{d}$$
$$C(2,2) = \frac{(1+v_y)(1+\Delta u_x)}{d} - \frac{v_x \Delta u_y}{d}$$
$$C(2,3) = v_0 + \frac{(1+v_y)(\Delta u_0 \Delta v_x - \Delta u_x \Delta v_0 - \Delta v_0)}{d} + \frac{v_x(\Delta u_y \Delta v_0 - \Delta u_0 \Delta v_y - \Delta u_0)}{d}$$

where $d = 1 + \Delta u_x - \Delta u_y \Delta v_x + \Delta v_y + \Delta u_x \Delta v_y$.

Matrix $\mathbf{C}$ can be simplified by neglecting all products of incremental factors. Furthermore, by noting that u_x, u_y, v_x and v_y are more than one order of magnitude smaller than displacements, all mixed products (e.g. $u_y \Delta u_x$, $u_x \Delta u_y$, ...) can also be neglected. The first order approximation of $\mathbf{C}$ thus is

$$\hat{C}(1,1) = \frac{1+u_x+\Delta v_y}{1+\Delta u_x+\Delta v_y} \qquad \hat{C}(1,2) = \frac{u_y-\Delta u_y}{1+\Delta u_x+\Delta v_y} \qquad \hat{C}(1,3) = u_0 - \frac{\Delta u_0}{1+\Delta u_x+\Delta v_y}$$
$$\hat{C}(2,1) = \frac{v_x-\Delta v_x}{1+\Delta u_x+\Delta v_y} \qquad \hat{C}(2,2) = \frac{1+v_y+\Delta u_x}{1+\Delta u_x+\Delta v_y} \qquad \hat{C}(2,3) = v_0 + \frac{\Delta v_0}{1+\Delta u_x+\Delta v_y}$$

where $1 + \Delta u_x + \Delta v_y$ is the first order approximation of d.

$\hat{C}(1,1)$ and $\hat{C}(2,2)$ can be further simplified by summing and subtracting to the numerator Δu_x and Δv_y respectively, thus obtaining

$$\hat{C}(1,1) = \frac{1+u_x+\Delta v_y+\Delta u_x-\Delta u_x}{1+\Delta u_x+\Delta v_y} = 1 + \frac{u_x-\Delta u_x}{1+\Delta u_x+\Delta v_y}$$

and

$$\hat{C}(2,2) = \frac{1+v_y+\Delta u_x+\Delta v_y-\Delta v_y}{1+\Delta u_x+\Delta v_y} = 1 + \frac{v_y-\Delta v_y}{1+\Delta u_x+\Delta v_y}$$

Finally, we can observe that the factors Δu_x and Δv_y in the denominator of all the matrix entries are very small and go to zero with iterations, so the denominator is almost unitary, i.e. the (IC-GN) matrix $\mathbf{C}$ is identical to the BS-GN matrix (9) to the first order.

5. Experimental Validation

To validate the new formulation, we processed one of the experimental datasets related to the polynomial bias analysis described in [19] with the new algorithm. As the experimental apparatus and its calibration are fully detailed in the aforementioned paper, we will only summarize the relevant data in this article: each data set consists of 43 images acquired during a rigid body translation test. The experiment simulates real-use conditions, so the images were acquired using a telecentric lens (Edmund Optics 0.9× Large-Format Telecentric Lens). Consequently, the displacement corresponding to one pixel of our camera (AVT Pike F-421

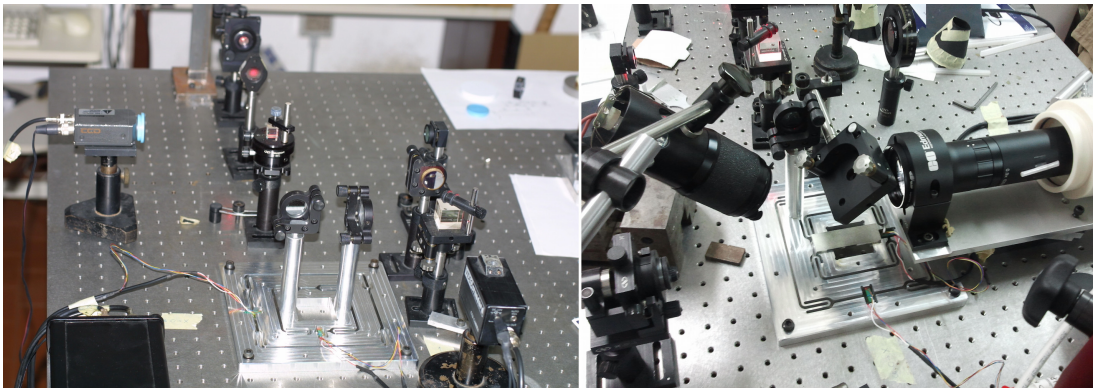


Figure 3: Experimental setup. Left: during calibration; note the two Twyman–Green Interferometers used to accurately measure displacements. Right: experimental configuration during DIC tests. One of the interferometers has been removed to allow the installation of a 45-degree mirror in front of the telecentric lens. The second interferometer is still installed, but is not used during the acquisition: displacements are recorded using the strain gauge bridges installed on piezo actuators.

FireWire-B 1.2 CCD camera; 2048 × 2048 pixels; pixel size 7.4 μm) is less than 10 μm. To accurately control displacements, a monolithic translation stage actuated by two low-voltage piezo crystals (ThorLabs PZS001, nominal displacement 11.6 ± 2 μm at 100 V) was built. The PZTs are instrumented using strain gauges (full-bridge configuration) to allow closed-loop control of the system.

Figure 3 shows the experimental setup during calibration (left) and during actual test (right). To correlate the strain gauge bridge signal with the actual displacements, two Twynam-Green interferometers were built taking care to install one of the mirrors on the translation stage. Thus, an accurate calibration was performed by monitoring the phase signal[5]. Figure 3-right shows the experimental setup during analysis. To reduce vibrations, the camera was installed horizontally on the optical bench: the specimen is thus imaged by a 45 degree tilted mirror. For installation, one of the two interferometers (no longer required) was removed.

Figure 4 shows the results of the analysis. Each curve plots the difference between the mean displacement value and the strain-gauge measured translation. Specifically, we sampled the image surface over a regular grid, taking care to obtain independent results (i.e. the sampling step is larger than a subset, so that each subset does not share pixels with its neighbors). For each image, we then computed the mean and standard deviation of the displacements. Finally, we plotted the difference between the normalized displacement (as acquired by strain-gauge signals) and the mean of the DIC-estimated displacements. In order to perform the normalization, we had to identify pixel size. This should be 8.22 μm (the nominal pixel size is 7.4 μm and the (nominal) primary magnification of the telecentric lens is 0.9×), but the actual size depends on the alignment of the lens and the actual primary magnification. Instead of carrying out a difficult experimental analysis, we simply examined the error curves associated with different interpolation polynomials. As none of them shows bias for displacement equals to $n/2$ pixel, it suffices to look for their intersection points: the second corresponds to a shift of one-pixel.

Analysis of fig. 4 shows that the BS-GN method behaves in the same way as the IC-GN. It is well known that the IC-GN algorithm is less affected by polynomial bias than FA-GN because

[5] For measurement, strain gauges are much more convenient than interferometers because they do not require any post-processing (apart from the scale factor) and provide absolute data.

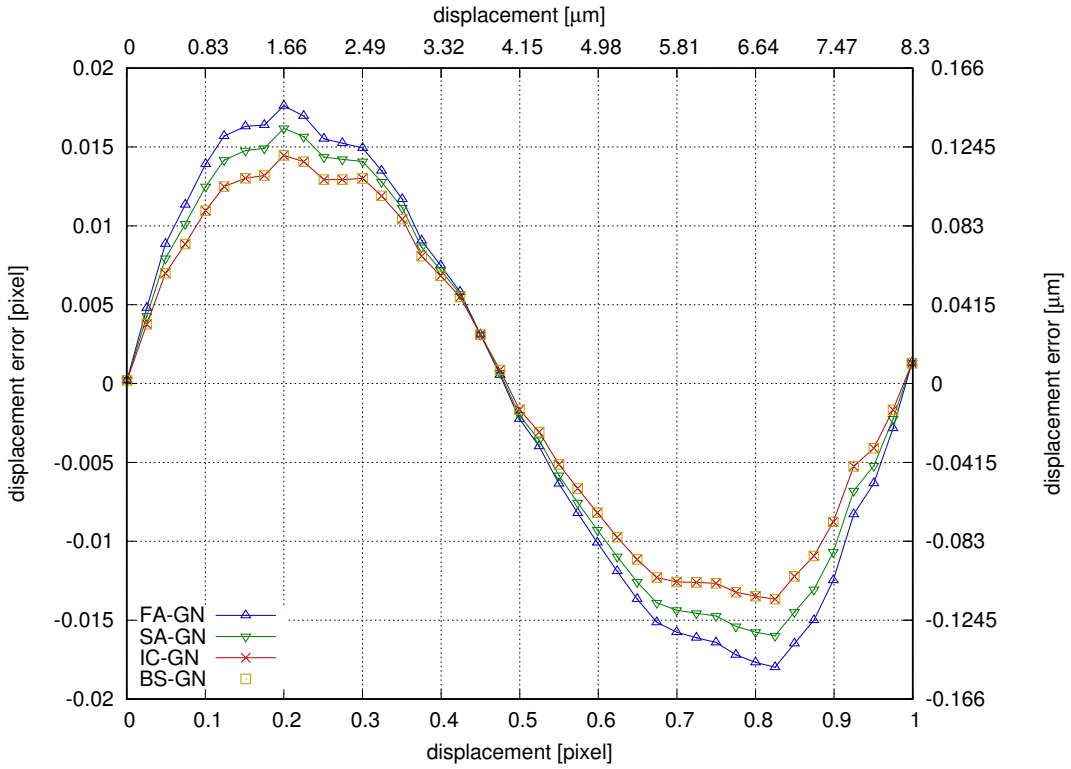


Figure 4: Bias of the c_4 polynomial. The Forward-Additive Gauss-Newton method (FA-GN) shows the largest bias whereas the Inverse-Compositional (IC-GN) and Backward-Subtractive (BS-GN) methods the smallest. The Symmetric-Additive approach stays in the middle.

the intensity derivatives are calculated at integer locations. But the Backward-Subtractive algorithm also uses pre-computed image intensity derivatives estimated at integer locations, so, we expect a similar performance. Indeed, the results of the two approaches are so similar that they appear as a single curve in the figure. The Forward additive shows a somewhat larger bias and the symmetric approach falls in the middle.

Figure 5 shows the results of the analysis of sample 14 of the DIC Challenge dataset. The 2D–DIC Challenge initiative [20] supplies a set of sample images and a set of analysis guidelines for standardized reporting of 2D–DIC results. Of all the datasets, Sample 14 is becoming the de facto standard for testing DIC algorithms and for this reason we show here the results of the BS-GN algorithm when dealing with this data set.

We analyzed the L_1, L_3 and L_5 load cases using a subset size of 21×21 pixels, linear shape functions and the BS-GN formulation. The image surface was sampled every two pixels and radiosity compensation was enabled to mimic real-use conditions (actually, radiosity compensation is not required as the images are numerically generated, so there is no illumination variation). Figure 5-top shows the mean displacement versus the x-location of the three load cases. As the displacements are only along the x axis, it is possible to statistically analyze each column of the image to extract the mean and standard deviation of the estimated displacements. The translucent areas around the solid lines correspond to ± 1 standard deviation error. We have also reported the expected displacements: apart from a few local errors, a fairly good agreement with the expected results was observed. Figure 5-bottom shows the density plot and contour lines associated withthe L_5 load case to visually support the analysis: the blue regions correspond to maximum negative displacements while the yellow areas correspond to maximum positive values

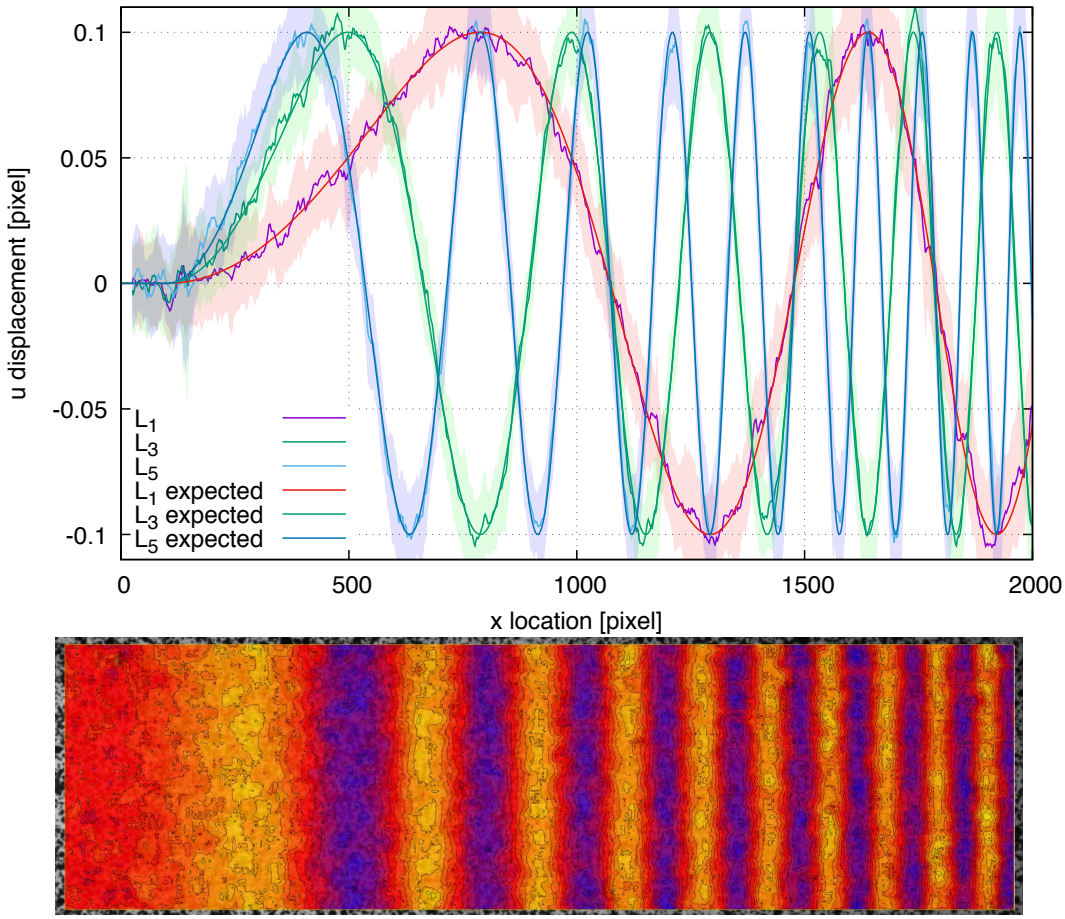


Figure 5: DIC Challenge Sample 14. Top: mean displacements related to load levels L_1, L_3 and L_5. The translucent regions correspond to ± 1 standard deviation. Bottom: density plot of u displacements related to step L_5.

(respectively -0.1 and $+0.1$ pixel).

6. Discussion and Conclusions

This paper discusses a modification of the Inverse Compositional DIC formulation. In particular, the inversion and composition steps are replaced by a simple subtraction. Since the Hessian matrix is evaluated in the template image at integer locations, it can be pre-evaluated at the beginning of the computation exactly as in the IC-GN method. For the same reason, the proposed approach is not affected by noise-bias when appropriate differential operators are used [21]. Furthermore, the inversion and matrix multiplications are replaced by a simple subtraction, so the resulting algorithm is theoretically somewhat faster[6].

A significant advantage of the proposed formulation is the fact that it removes the constraint of using invertible shape functions. Although a parabolic formulation has recently been proposed [18], there is no solution for higher order shape functions, making it impossible to correctly describe complex strain gradient regions. Finally, although the displacement field should always be invertible when affine shape functions are used [6], the matrix may be ill-conditioned and

[6] The speed gain is almost unnoticeable if linear shape functions are used, but may become significant if higher order shape functions are used.

introduce significant numerical errors. This problem obviously does not exist in the proposed formulation.

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] A. Del Bimbo, P. Nesi, and J.L.C. Sanz. Analysis of optical flow constraints. *IEEE Trans. Image Process.*, 4(4):460–469, Apr 1995. ISSN 1057-7149. doi: 10.1109/83.370674.

[3] Berthold Horn, Berthold Klaus, and Paul Horn. *Robot vision*. MIT press, 1986.

[4] RB Lehoucq, PL Reu, and DZ Turner. The effect of the ill-posed problem on quantitative error assessment in digital image correlation. *Experimental Mechanics*, pages 1–13, 2017.

[5] 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.

[6] W Tong. Formulation of lucas–kanade digital image correlation algorithms for non-contact deformation measurements: a review. *Strain*, 49(4):313–334, 2013.

[7] B. Pan, H. Xie, and Z. Wang. Equivalence of digital image correlation criteria for pattern matching. *Applied Optics*, 49(28):5501–5509, 2010.

[8] André-Louis Cholesky. Sur la résolution numérique des systèmes d'équations linéaires. *Bulletin de la Sabix. Société des amis de la Bibliothèque et de l'Histoire de l'École polytechnique*, (39):81–95, 2005.

[9] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes. The Art of Scientific Computing*. Cambridge University Press, 3rd edition, 2007. ISBN 0-521-88068-8.

[10] J Neggers, B Blaysat, JPM Hoefnagels, and MGD Geers. On image gradients in digital image correlation. *International Journal for Numerical Methods in Engineering*, 105(4): 243–260, 2016.

[11] Zhengyou Zhang. Motion and structure from two perspective views: from essential parameters to euclidean motion through the fundamental matrix. *JOSA A*, 14(11):2938–2950, 1997.

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

[13] 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.

[14] 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.

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

[16] Heung-Yeung Shum and Richard Szeliski. Construction of panoramic image mosaics with global and local alignment. *International Journal of Computer Vision*, 48(2):151–152, 2002.

[17] Lloyd N Trefethen and David Bau III. *Numerical linear algebra*, volume 50. Siam, 1997.

[18] Ruixiang Bai, Hao Jiang, Zhenkun Lei, and Weikang Li. A novel 2nd-order shape function based digital image correlation method for large deformation measurements. *Optics and Lasers in Engineering*, 90:48–58, 2017.

[19] 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.

[20] Phillip L Reu, Eveline Toussaint, Elizabeth Jones, Hugh A Bruck, Mark Iadicola, Ruben Balcaen, Daniel Z Turner, Thorsten Siebert, Pascal Lava, and M Simonsen. Dic challenge: developing images and guidelines for evaluating accuracy and resolution of 2d analyses. *Experimental Mechanics*, pages 1–33, 2017. URL `http://sem.org/dicchallenge/`.

[21] Xinxing Shao, Xiangjun Dai, and Xiaoyuan He. Noise robustness and parallel computation of the inverse compositional gauss–newton algorithm in digital image correlation. *Optics and Lasers in Engineering*, 71:9–19, 2015.