

Determination of In-Plane and Off-Plane Surface Displacements with Grids Virtually Applied to Digital Images

Valerio Lux, Emanuele Marotta and Pietro Salvini

Abstract Digital image devices have experienced an enormous increment in their capabilities, associated with a significant reduction in the economic effort. In particular, the increasing number of pixel made available for each picture allows developing software that is able to perform precise surface characterizations. In the present chapter the interest is oriented into two directions. The first one concerns detecting the geometric features of surfaces through digital image comparison. The method does not require stereo image processing but it is based on a single camera vision. The base of this first part of the work regards the displacements of a grid virtually applied on the surface. To this goal the real printed grid case is firstly discussed. The grid virtually attached to the pictures identifies a finite element mesh associated to the comparing images. The second part aims to evaluate surface strains experienced on the specimen surface. The algorithm performs the analysis of the two comparing images, before and after the application of loads. Two different strategies are proposed: a partial grouping of pixels by equation averaging; the use of Hu's invariants applied to sub-images.

Keywords Strain measurements · Digital image correlation · Finite elements

1 Introduction

Image interpretation and processing is a fundamental analysis in most of medical applications [5]. The main objective is to help the analyst in the interpretation of the results while introducing quantitative data. In some cases, the attention is focused on surface detection, while in other contexts, the observation is dedicated to morphological changes or surface extension (strain or deformations).

V. Lux · E. Marotta · P. Salvini (✉)
Department of Industrial Engineering, Università di Tor Vergata, Roma, Italy
e-mail: salvini@uniroma2.it

Considering rigid body detection and surface out of plane measurements, this item has been extensively studied by many researchers [11]. In many cases the objective is directed towards the recognition of objects, whatever is their spatial positioning and minimal surface deformation. This is often accomplished by methods that make use of patterns of key points [9].

For the reconstruction of surface deformations many efforts have been done by researchers, in both 2D and 3D approach, some of these methods have been compared in [14]. The general achievements show that many benefits are gained if the entire acquired image is simultaneously considered; this is to say non-considering each subset individually. In this optics, some smoothing can be achieved by B-spline regularization [3] or by finite element formulation for the acquisition of the deformation field [7] by DIC.

In the present chapter the attention is firstly devoted on the capability to extract the geometric shapes of a surface, originally flat. This means that the interest is not only directed towards object recognition, but to new surface characteristics for the identification of its off-plane displacements. In the present case the tests consider the displacement of a sheet of paper, easily deformed in the out of plane orientation but experiencing minor surface strains. Within the paper, it is demonstrated that using an effective image, sufficiently variegated (such as a speckle image or any other non-uniform and non-periodical picture) is equivalent to a regularly meshed grid [1]. This equivalence is achieved by mapping the picture through a regular or non-regular mesh of quadrilateral sub-images (elements). These non-superimposed sub-images are managed as four node bilinear membrane elements, well known in finite element analysis. Each element of the grid maintains its peculiarity because it is characterized by a different color content and distribution (sub-image).

Another aspect, that makes use of the same image recognition methods, regards the capability to extrapolate the strains experienced by mating surfaces using non-invasive methods. The most promising methods involve photographic techniques, such as digital image correlation [4, 12–14]. Usually the strain and motion analysis procedures make use of a sequence of pictures that follows the whole strain progress [2]. In this chapter a method that requires only two images is proposed; the first is taken when no loads are applied, the second one when all loads and resulting deformations are settled. This method can thus be applied in principle when a limited number of images are available, as the initial and the final image only.

There exist 2-D digital Correlation techniques as well as 3-D ones [10]. Here we deal with 2-D technique which is based on the use of a single camera. Most of the techniques are based on sub-image correlations. This means that the local correlation imposed on a sub-image does not have an influence on all other correlations performed on sub-images far away from the previous one. In this chapter, according to former approaches [1, 3], we discuss a technique that solves the displacement fields as a whole, so that the continuity conditions is fulfilled in the whole processed image.

When consistent displacements are faced, the correlation techniques meet considerable difficulties to keep precision, as discussed in [12].

The displacement field solution generally needs the handling of a very high number of equations; this can even attain the order of the number of pixels recorded into the image. Therefore, a technique directed to reduce the number of equations while increasing the efficiency is discussed.

2 Out of Plane Surface Deformation

The geometric model adopted is the simple equivalent pinhole camera. According to this assumption, being f the distance between the pinhole and the image sensor, the following ratios can be written:

$$X' = -\frac{f}{Z}X; \quad Y' = -\frac{f}{Z}Y; \quad Z' = -f \tag{1}$$

The negative sign is generally changed by considering the image projected between the viewing point and the object (Fig. 1).

In the pinhole assumption, the complex dioptric lens system is substituted by an ideal single lens, infinitesimally thin, thus the optical system agrees with the following assumptions [15]:

- all parallel light rays are concentrated on the focus;
- no refraction is induced to all rays passing through the lens center;
- all non-centered rays are deviated in correspondence of the middle plane.

Other assumed hypotheses, adopted when dealing with digital images are:

the optical axis is perfectly orthogonal to the sensor plane and centered on it;
 the sensor gauge is organized by a two perfectly orthogonal cell disposition.

2.1 Identification of Grid Images

In this section we assume that the image is simply constituted by a regular square grid. In the next section the association of a grid to a general image is discussed. Figure 2 shows the projection of a simple square on the CCD plane that is reversed, as usually. The geometry projected on the plane is given by the four vectors connecting the observation point to the square corners.

Once the image is digitally acquired, each vector \mathbf{v} associated to a point is known in its direction, while its magnitude remains unknown. All unknowns are represented by the moduli associated to the respective vectors. According to this logic, all vectors \mathbf{U} are computed as differences of vectors \mathbf{V} (Fig. 2). In the following, the vectors \mathbf{V} will be addressed making use of respective unit vectors: $\mathbf{V} = m \cdot \mathbf{v}$.

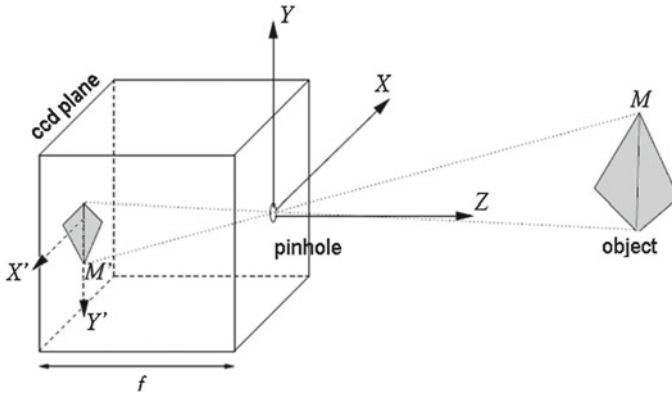


Fig. 1 Scheme of pinhole camera view

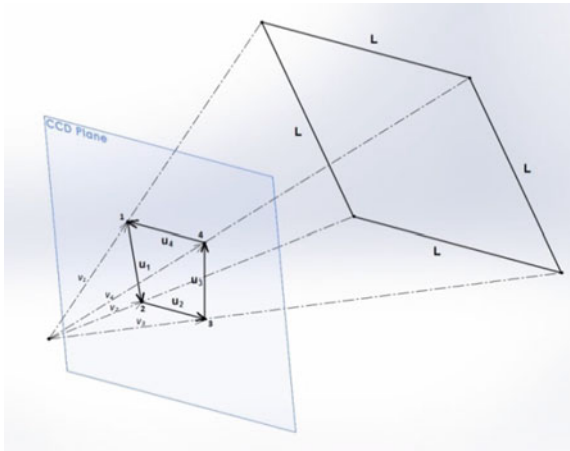


Fig. 2 Vectors identifying the positioning of element nodes

Making reference to the 3×3 grid represented in Fig. 3, the first element gives the following equations:

$$\begin{aligned}
 \mathbf{U}_1 &= \mathbf{V}_5 - \mathbf{V}_1 = m_5 \mathbf{v}_5 - m_1 \mathbf{v}_1 \\
 \mathbf{U}_2 &= \mathbf{V}_6 - \mathbf{V}_5 = m_6 \mathbf{v}_6 - m_5 \mathbf{v}_5 \\
 \mathbf{U}_3 &= \mathbf{V}_2 - \mathbf{V}_1 = m_2 \mathbf{v}_2 - m_1 \mathbf{v}_1 \\
 \mathbf{U}_4 &= \mathbf{V}_6 - \mathbf{V}_2 = m_6 \mathbf{v}_6 - m_2 \mathbf{v}_2
 \end{aligned}
 \tag{2}$$

Being the grid formed by equal squares, several conditions can be imposed to each of them - note that not all of them are independent - represented in Table 1.

Fig. 3 Nomenclature of a 3×3 grid

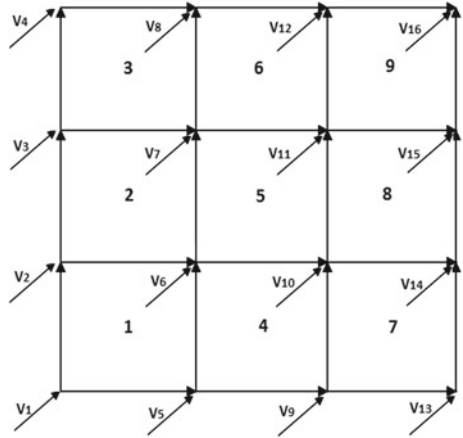


Table 1 Vector and scalar conditions for a square grid

Geom. condition	Vector eq.	Number of scalar eqs
$U_2 \parallel U_3$	$U_2 \times U_3 = 0$	3
$U_1 \parallel U_4$	$U_1 \times U_4 = 0$	3
$U_1 \perp U_2$	$U_1 \cdot U_2 = 0$	1
$U_2 \perp U_4$	$U_2 \cdot U_4 = 0$	1
$U_3 \perp U_4$	$U_3 \cdot U_4 = 0$	1
$U_1 \perp U_3$	$U_1 \cdot U_3 = 0$	1
$ U_1 = L$	$U_1 \cdot U_1 = L^2$	1
$ U_2 = L$	$U_2 \cdot U_2 = L^2$	1
$ U_3 = L$	$U_3 \cdot U_3 = L^2$	1
$ U_4 = L$	$U_4 \cdot U_4 = L^2$	1

The equations given in the previous Table 1 generates, using the modules as the unknowns, the following 14 equations (first two are vector equations):

$$\begin{aligned}
 m_1 m_5 (\mathbf{v}_5 \times \mathbf{v}_1) - m_1 m_6 (\mathbf{v}_6 \times \mathbf{v}_1) - m_2 m_5 (\mathbf{v}_5 \times \mathbf{v}_2) - m_2 m_6 (\mathbf{v}_6 \times \mathbf{v}_2) &= \mathbf{0} \\
 m_1 m_2 (\mathbf{v}_1 \times \mathbf{v}_2) - m_1 m_6 (\mathbf{v}_1 \times \mathbf{v}_6) - m_2 m_5 (\mathbf{v}_5 \times \mathbf{v}_2) + m_5 m_6 (\mathbf{v}_5 \times \mathbf{v}_6) &= \mathbf{0}
 \end{aligned} \tag{3}$$

$$\begin{aligned}
 m_1 m_5 (\mathbf{v}_1 \cdot \mathbf{v}_2) - m_1 m_6 (\mathbf{v}_1 \cdot \mathbf{v}_6) - m_5 m_5 (\mathbf{v}_5 \cdot \mathbf{v}_5) + m_5 m_6 (\mathbf{v}_5 \cdot \mathbf{v}_6) &= 0 \\
 -m_2 m_6 (\mathbf{v}_2 \cdot \mathbf{v}_6) + m_2 m_5 (\mathbf{v}_2 \cdot \mathbf{v}_5) - m_5 m_6 (\mathbf{v}_5 \cdot \mathbf{v}_6) + m_6 m_6 (\mathbf{v}_6 \cdot \mathbf{v}_6) &= 0 \\
 m_1 m_2 (\mathbf{v}_1 \cdot \mathbf{v}_2) - m_1 m_6 (\mathbf{v}_1 \cdot \mathbf{v}_6) - m_2 m_2 (\mathbf{v}_2 \cdot \mathbf{v}_2) + m_2 m_6 (\mathbf{v}_2 \cdot \mathbf{v}_6) &= 0 \\
 m_1 m_1 (\mathbf{v}_1 \cdot \mathbf{v}_1) - m_1 m_2 (\mathbf{v}_1 \cdot \mathbf{v}_2) - m_1 m_5 (\mathbf{v}_1 \cdot \mathbf{v}_5) + m_2 m_5 (\mathbf{v}_2 \cdot \mathbf{v}_5) &= 0
 \end{aligned} \tag{4}$$

Table 2 List of unknowns produced by a single element

Unknowns			
1	m_1^2	6	m_2m_6
2	m_1m_2	7	m_2m_5
3	m_1m_6	8	m_5^2
4	m_1m_5	9	m_5m_6
5	m_2^2	10	m_6^2

$$\begin{aligned}
m_1^2 \sum_{i=1}^3 v_{1(i)}^2 - 2m_1m_3 \sum_{i=1}^3 (v_{1(i)}v_{5(i)})^2 + m_5^2 \sum_{i=1}^3 v_{5(i)}^2 &= L^2 \\
m_5^2 \sum_{i=1}^3 v_{5(i)}^2 - 2m_5m_6 \sum_{i=1}^3 (v_{5(i)}v_{6(i)})^2 + m_6^2 \sum_{i=1}^3 v_{6(i)}^2 &= L^2 \\
m_2^2 \sum_{i=1}^3 v_{2(i)}^2 - 2m_2m_1 \sum_{i=1}^3 (v_{2(i)}v_{1(i)})^2 + m_1^2 \sum_{i=1}^3 v_{1(i)}^2 &= L^2 \\
m_6^2 \sum_{i=1}^3 v_{6(i)}^2 - 2m_6m_2 \sum_{i=1}^3 (v_{6(i)}v_{2(i)})^2 + m_2^2 \sum_{i=1}^3 v_{2(i)}^2 &= L^2
\end{aligned} \tag{5}$$

All above equations give a non-linear (quadratic) system of equations where the unknowns are vector magnitudes. If one considers all possible combinations of products of unknowns as unknowns themselves, they turn to be 10 for a single element, with 14 equations each. As an example, for the element n. 1 of Fig. 3, the 10 unknowns are listed in Table 2.

The full system of equations is therefore over-determined and the solution can be found solving all the quadratic unknowns involved. After the full solution, each vector magnitude is computed by the root square of the quadratic unknowns. Furthermore, the mixed product of the unknowns can be used to check the accuracy of the solution gained.

2.2 Virtual Image Embedded on a Picture

If an image is present on a surface, this can be associated to a virtual grid. The point is to guarantee that the grid follows the changes of the image, due to movements of the surface that can be considered as a combination of rigid and deformable displacements.

This can be accomplished by considering the grid as a mesh of bilinear finite elements, whose movements guarantee the continuity of the surface. Each element contains a part of the initially flat image; this information is maintained in a natural coordinate system as shown in Fig. 4.

Such reference approach is particularly suitable to compare elements that are initially irregular or become irregular due to large displacements on the image. Therefore, each sub-image is interpolated though a cubic spline approach. By this

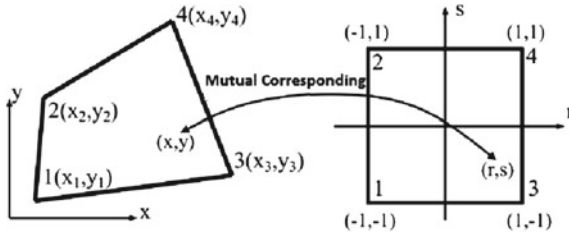


Fig. 4 Physical/Natural coordinate systems

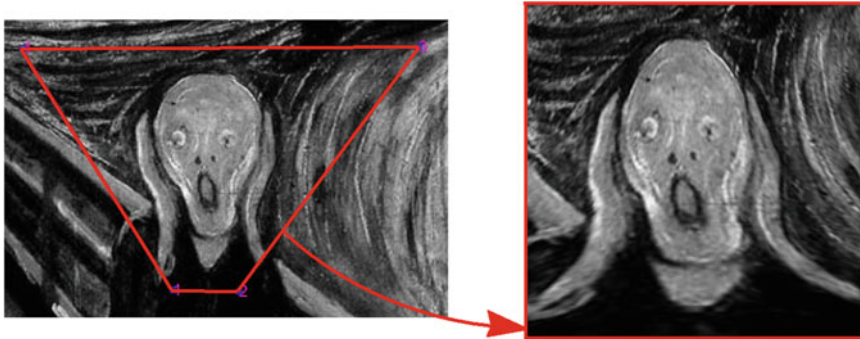


Fig. 5 Change from physical to natural coordinates

procedure, each element is always square-represented and keeps the same image content. In Fig. 5 is shown an example of how interpolation deforms the image.

2.3 Results on Out of Plane Deformation

The data here presented, apart the very next subsection, refers to all effective pictures taken with a focal distance equal to 29 mm, corresponding to a printed paper positioned at 1500 mm from the ideal lens center. All images have been obtained with an aperture equal to 1/8 to increase overall focus depth.

2.3.1 Test of the Procedure on Exact Grids

The theoretical correctness of the procedure presented has first of all been investigated through the application on a simulated grid (no pictures does effectively exist) that has been deformed applying one or two finite curvatures. We can see that the reconstruction is perfectly accurate (no digital error on pixel definitions is present since pixel positions are recorded with 12 digit precision) only if the local orientation

Table 3 Noise effect on accuracy for various curvatures

	No noise (%)	Added noise (%)
Plane grid	$\sim 1e-10$	4.06
Single curvature	$\sim 1e-10$	4.87
Double curvature	23.13	24.21
D.curv. refined el.	13.61	15.93

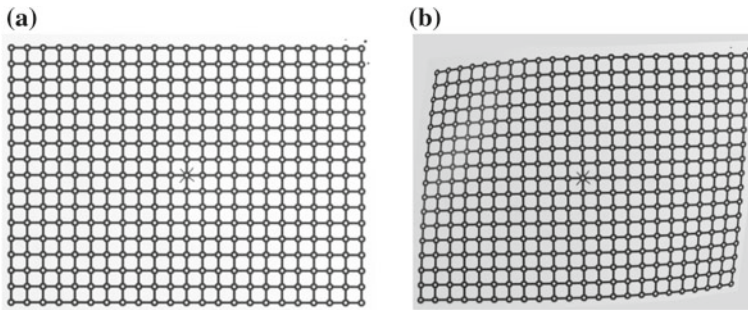


Fig. 6 Digital pictures of the printed grid, **a** before and **b** after off-plane deformation

of the grid causes a single curvature change ($R_c = 1430$ mm). It is interesting to highlight that the application of a small noise (1 % of the diagonal length of a single element) reduces considerably the precision if the results are very accurate, but does not appreciably change the behavior if discrepancies are already encountered when data are not affected by noise.

From the above reasons, it is clear that the assumption of square grid to maintain its shape is very strong, difficult to obtain when double curvature are present. This means that the size of the elements of the grid should be taken as small as the curvature increases. As a matter of fact, the last row in Table 3 shows much better results in this case, as expected (double curvature keeping the same equivalent value than single curvature case).

2.3.2 Tests on Pictures of Printed Grids

The grid is printed on a sheet of paper that is first photographed in a plane orthogonal to the focal axis, and the second picture considers the sheet deformed in various ways, such as the one shown in Fig. 6.

Four cases are here presented; (i) it concerns a $5 - 10 - 15^\circ$ rigid rotation of the paper on a vertical axis as to generate a prospective view; (ii) it regards a simple half-fold oriented as the vertical axis in the center, and folded at a corner; (iii) the paper leaned on a cylinder (diameter = 450 mm) with a vertical axis; (iv) the paper applied on the same cylinder with the sheet base inclined of 30° .

In Fig. 7 two operating ways are compared: the black lines show the results when the over-determined solution is performed on all squares at the same time, the blue

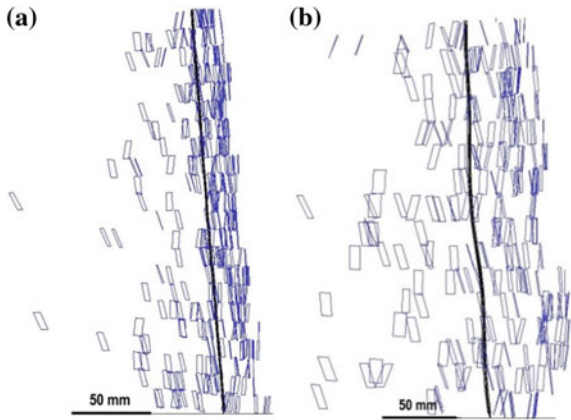


Fig. 7 Comparison of overall result and square dispersion for; **a** printed grid, **b** virtual grid on speckle image

Table 4 Angle identified when varying the point of view

Angle (°)	Grid method results (°)	Speckle method	
		16 × 16 el. (°)	37 × 37 el. (°)
5	4.64	3.96	4.50
10	9.72	8.23	8.69
15	14.86	9.64	11.68

lines consider the solution performed at each square, individually. The whole solution is much better than the second, since pixel errors compensate.

Table 4 shows the errors introduced when varying sheet inclination.

The results show that a rigid displacement on one of the grid axes can be managed by the method in the average by both images (grid or speckle above discussed) (Fig. 8).

The two cases of folds considered (ii) reveal a particular behavior of the algorithm. As a matter of fact, the mean square method tends to compensate the errors so that there is the tendency to keep flat in one direction. Referring to Fig. 9a and b the dispersion is illustrated when the fold is located in the center or in the corner, respectively. The maximum determined fold displacement respects the values imposed in the test within a 2% error. Figure 10b results seem to be erroneously similar to the previous case, as the top views show. It is clear that the identification algorithm is affected by eventual curvatures non-aligned with the grid. In practice, since the algorithm tends to maintain the overall length at each element, it encounters some difficulties while managing elements that change all side-length due to curvatures imposed.

The results performed on cylindrical surfaces confirm the deficiencies previously indicated on double curvature grids. As a matter of fact the comparison between

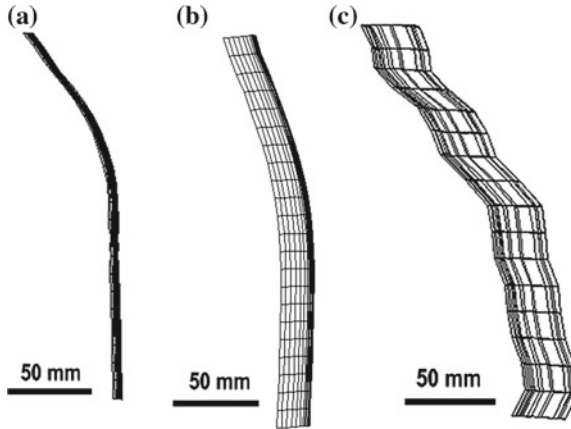


Fig. 8 Top view of the profiles for case (ii): **a** vertical fold **b** corner fold, **c** vertical fold for speckle image

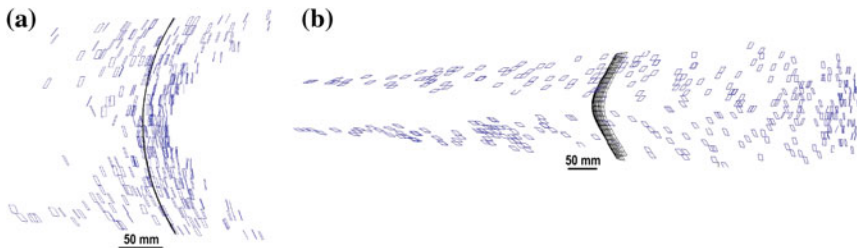


Fig. 9 Dispersion of individually identified elements for: **a** vertical cylinder, **b** inclined cylinder

cases (a) and (b) in Fig. 9 shows a much more evident dispersion of single computed squares (blue) when the sheet base is inclined toward horizon—relative full picture is visible in Fig. 7b. The global results (black lines) of case (a) are quite accurate (error in radius lower than 1 %) while case (b) provides erroneous results.

2.3.3 Tests on a Printed Speckle Picture

Indeed, the application of a virtual grid introduces some additional errors by respect to the printed grid. These errors amplify the deficiencies already evidenced before. The algorithm used to detect the displacement of the virtual grid by means of speckle deformed image is presented in the next session. The convergence is achieved by means of minimum square error search. For example, one can compare the results given in Figs. 10 and 11a and b, respectively. Case (a) identifies the virtual grid imposed on the image to be taken as reference; case (b) in Fig. 10 shows good identification that can also be seen in Fig. 7b through square dispersion. On the

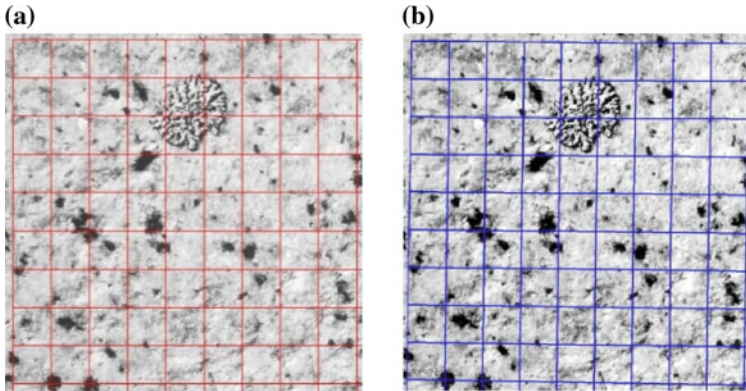


Fig. 10 Virtual node locations before and after 5° inclination

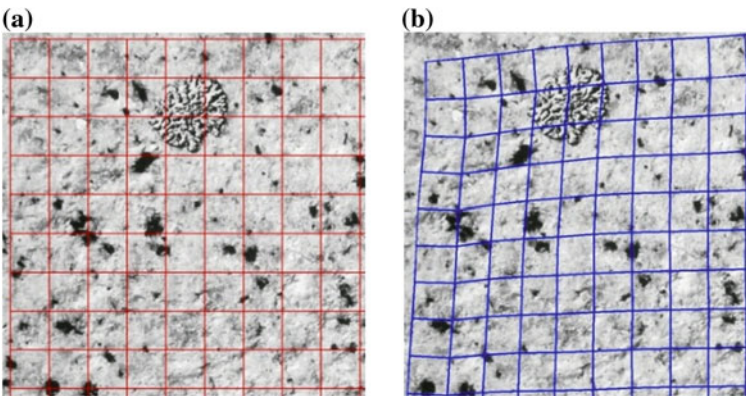


Fig. 11 Virtual node locations before and after application on inclined cylinder

contrary, one can see that the corner correspondence on Fig. 11a, b is very poor, particularly at the left top corner where the deformations are the highest, according to results presented in Fig. 9b.

2.3.4 Discussion on Out of Plane Deformations

It is evident that algorithm proposed here shows some difficulties when applied to structures that deforms with effective double curvatures. As a matter of fact, when double curvatures are present, the side lengths cannot be computed accurately by simple node distances. A better computation should accounts of effective surface distance through an iterative procedure that takes into account of geometry on curved surfaces. From another point of view, the advantage of the method is that no regularization conditions are required to the identifying surface, so that no simplified

shape is accounted for the surface out of plane deformation. An idea of the solution accuracy can be reached from the dispersion of the results when computation regards each square element individually. When the dispersion is high the global results is correspondently worse. Use of speckle image instead of a printed grid is possible and a theoretical increment of information is available. However, the differential method, discussed hereinafter, requires a limitation of the displacements introduced in the image to keep consistency. When the strain are less than 50 % of the virtual grid nodes moves correctly, for higher values accuracy problems become evident.

3 Differential Method

The deformation field is calculated by comparing the original image and the deformed one. In this work a global approach is followed [2]. The problem consists in the minimization of the error functional defined by the following formula:

$$E = \sqrt{\sum_i [I_d(x_i, y_i) - I_u(x_i, y_i)]^2} \quad (6)$$

where $I_u I_d$ represent the images before and after deformation. x_i, y_i are the i -th pixel coordinates into the images. The summation comprehends all pixels. To each element a sub-image is associated. At each sub-image (corresponding to a single finite element) the internal spatial distribution is based on the displacements of the four nodes bordering the element. The solution allows finding the node locations that make it possible to overlap the undeformed image onto the deformed one. Once the image is divided into sub-images, it is possible to write the formula (6):

$$E = \sqrt{\sum_j \|\mathbf{S}_j(\mathbf{x}) - \mathbf{S}_{j0}(\mathbf{x}_0)\|^2} \quad (7)$$

where $\mathbf{S}_j, \mathbf{S}_{j0}$ are the j -th sub-image, while \mathbf{x} and \mathbf{x}_0 represent the final and initial nodal coordinates, respectively. The rate of change between \mathbf{x} and \mathbf{x}_0 is non-linear; for this reason it is not possible to directly find the solution, but an iterative procedure is needed. The algorithm consists in the linearization of the non-linear least squares problem. It is based on Taylor series expansion, truncated to its first order, of the sub-image when varying the generic nodal coordinate:

$$\mathbf{S}_j(X') = \mathbf{S}_0(X_0) + \sum_k \frac{\partial \mathbf{S}_0}{\partial x_k} (x_k - x_{k0}) \quad (8)$$

For the sake of clarity, the same is written in matrix formulation, evidencing the Jacobian matrix:

$$\begin{bmatrix} \frac{\partial \mathbf{S}_{1,0}}{\partial X_1} & \frac{\partial \mathbf{S}_{1,0}}{\partial X_2} & \cdots & \frac{\partial \mathbf{S}_{1,0}}{\partial X_M} \\ \frac{\partial \mathbf{S}_{2,0}}{\partial X_1} & \frac{\partial \mathbf{S}_{2,0}}{\partial X_2} & \cdots & \frac{\partial \mathbf{S}_{2,0}}{\partial X_M} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \mathbf{S}_{N,0}}{\partial X_1} & \frac{\partial \mathbf{S}_{N,0}}{\partial X_2} & \cdots & \frac{\partial \mathbf{S}_{N,0}}{\partial X_M} \end{bmatrix} \cdot \begin{bmatrix} x_1 - x'_1 \\ x_2 - x'_2 \\ \vdots \\ x_M - x'_M \end{bmatrix} = \begin{bmatrix} \mathbf{S}_{1,0} - \mathbf{S}_{1,j'} \\ \mathbf{S}_{2,0} - \mathbf{S}_{2,j'} \\ \vdots \\ \mathbf{S}_{N,0} - \mathbf{S}_{N,j'} \end{bmatrix} \quad (9)$$

or the equivalent $\mathbf{J} \cdot \Delta \mathbf{x} = \Delta \mathbf{s}$. Each term of the Jacobian matrix is composed by a number of elements that coincides with the number of pixels contained in the sub-image. Therefore, the matrix shows considerable dimensions. The computation is based on centered first derivate, evaluated by considering the pixels around each node. For this reason, the Jacobian matrix of the reference image is computed by respect to all possible node displacements. As mentioned before, the Jacobian matrix is composed of partial derivatives. Each matrix derivative is calculated by the four central point derivatives by respect to the unknowns as:

$$\frac{\partial \mathbf{F}(x)}{\partial x} = \frac{1}{12h} [\mathbf{F}(x - 2h) - 8\mathbf{F}(x - h) + 8\mathbf{F}(x + h) - \mathbf{F}(x + 8h)] \quad (10)$$

where h denotes the discretization step. For example, the term $\mathbf{F}(x-2h)$ represents the sub-image when a node is moved back $2h$. The choice of the parameter h is crucial. The solution requires the minimization of the error (7) through an iterative procedure. As a matter of fact the inversion of the Jacobian matrix is required (9):

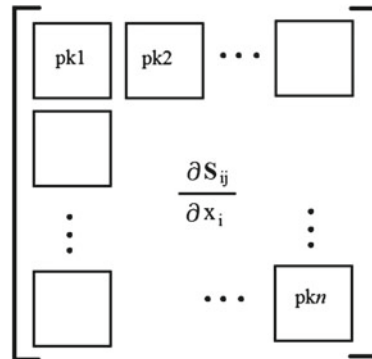
$$\Delta \mathbf{x} = \mathbf{J}^{-1} \cdot \Delta \mathbf{s} \quad (11)$$

The Jacobian matrix is a sensitivity matrix; however, its costly inversion must be performed just once. When the displacements of all nodes have been computed, it is easy to gain the internal strains, known by means of the nodal displacements of each element. This is made possible through the pre-multiplication of the vector of element nodal displacements by the matrix \mathbf{B} (obtain by appropriate derivative of Q4 shape functions) [16] if first order approximation is accepted, or more sophisticated expressions if first order simplification is not applicable.

3.1 Convergence Enhancement

In this section, the optimal choice of the number of internal points, to achieve a quick convergence of the results, is discussed. A criterion to reduce the equations and to speed-up the solution process is introduced. As an example, for a high-definition 3000×4000 pixel image, meshed through almost 10000 elements having eight degrees of freedom each, a 240 billion of equations results. In this chapter two possible techniques have been developed; the first considers possible grouping of pixels belonging to confined areas, the second one makes use of Hu's invariants to discriminate each sub-image.

Fig. 12 Grouping technique representation



3.1.1 Method I: Grouping Technique

The technique of grouping is helpful to reduce the number of equations to solve. As cited before, the Jacobian presents a considerable dimension; in fact each term of the matrix is a partial derivative of the sub-image. The grouping method introduced in this chapter, handles the single partial derivative of the sub-image. These are divided into sub-areas, and a sum of the values inside them is considered, as shown in Fig. 12.

By this way, the number of equations is reduced, and the iterative calculation speeds up. The size of the grouping is important, and must be wisely chosen. A considerable grouping amount is required for saving computational time, the data information, is reduced making it difficult to gain convergence in the iteration process.

3.1.2 Method II: Hu’s Invariants

Here we refer to moments as scalar quantities able to characterize a scalar field and possibly to point out some significant features. In statistics moments are widely used to describe the shape of a probability density function; in classic rigid-body mechanics to account of the mass distribution of a body, forming the inertia tensor. In mathematical terms, the moments are projections of a function onto a polynomial basis [6, 8]. General moments M_{pq} of an image $I(x,y)$, where p, q are non-negative integers and $r = p + q$ is called the order of the moment, are defined as:

$$M_{pq} = \iint p_{pq}(x, y) \cdot I(x, y) \cdot dx \cdot dy \tag{12}$$

Where $p_{00}(x, y), p_{10}(x, y), \dots, p_{pq}(x, y)$, are polynomial basis functions defined in the domain. In this chapter moments of the the image are used, consequently the function $I(x,y)$ is an image characterized by two coordinates x,y and a color value (e.g. RGB uses three values between 0 and 255 each).

A geometric moment of a discretized image is defined as a moment having the standard power basis $p_{pq}(x, y) = x^p y^q$. Therefore, it results:

$$m_{pq} = \sum_x \sum_y x^p \cdot y^q \cdot I(x, y) \tag{13}$$

Moreover Hu’s invariants are built up through a combination of geometrical moments that show the characteristics of non-changing their values when some geometrical transformations are applied. Hu’s invariants are originally seven, but other invariants could be computed. However, higher image invariants would increase considerably their magnitude, so that they cannot be managed together with the first seven.

Only the first invariant moment has an intuitive meaning: polar moment of inertia.

In this section Hu’s invariants are used to characterize each sub-image. Even using this technique, the solution requires the inversion of the Jacobian matrix; in this case the Jacobian is not calculated through image differences, but differences on Hu’s invariants. The image is meshed into sub-images by a grid; each sub-image is represented by its set of Hu’s invariant moments. Hu’s invariants are seven, but here, to increase the discriminant power, the computed invariants are doubled: they are computed on the sub-image itself and on its negative.

The invariant moments are insensitive to translations, rotations and scaling transforms. In an index compact notation they are:

$$\begin{aligned} \varphi_1 &= m_{20} + m_{02} \\ \varphi_2 &= (m_{20} - m_{02})^2 + 4m_{11}^2 \\ \varphi_3 &= (m_{30} - 3m_{12})^2 + (3m_{21} - m_{03})^2 \\ \varphi_4 &= (m_{30} + m_{12})^2 + (m_{21} - m_{03})^2 \\ \varphi_5 &= (m_{30} - 3m_{12})(m_{30} + m_{12}) [(m_{30} + m_{12})^2 - 3(m_{21} + m_{03})^2] + \\ &\quad + (3m_{21} - m_{03})(m_{21} + m_{03}) [3(m_{30} + m_{12})^2 - (m_{21} + m_{03})^2] \\ \varphi_6 &= (m_{20} - m_{02}) [(m_{30} + m_{12})^2 - (m_{21} + m_{03})^2] + \\ &\quad + 4m_{11}(m_{30} + m_{12})(m_{21} + m_{03}) \\ \varphi_7 &= (3m_{21} - m_{03})(m_{30} + m_{12}) [(m_{30} + m_{12})^2 - 3(m_{21} + m_{03})^2] + \\ &\quad - (m_{30} + 3m_{12})(m_{21} + m_{03}) [3(m_{30} + m_{12})^2 - (m_{21} + m_{03})^2] \end{aligned} \tag{14}$$

3.2 Natural Coordinate System Applied on Elements

Both methods require the image comparison to minimize the error. To this goal it is useful referring to the natural coordinate system used in the isoparametric element formulation [16].

Their characteristics are particularly suitable because, in the natural coordinate system, all the elements, as well as in the reference picture as in the deformed one, have the same shape and dimensions (a simple square as shown in Fig. 4).

For the isoparametric four-node element, all internal points are mapped through natural coordinates r,s in the following manner:

$$\begin{cases} x(r, s) = \frac{1}{4}(1-r)(1-s)x_1 + \frac{1}{4}(1-r)(1+s)x_2 + \\ \quad + \frac{1}{4}(1+r)(1-s)x_3 + \frac{1}{4}(1+r)(1+s)x_4 \\ y(r, s) = \frac{1}{4}(1-r)(1-s)y_1 + \frac{1}{4}(1-r)(1+s)y_2 + \\ \quad + \frac{1}{4}(1+r)(1-s)y_3 + \frac{1}{4}(1+r)(1+s)y_4 \end{cases} \quad (15)$$

As a matter of fact, one of the difficulties encountered in both methods regards the change of the edges during displacement, now overtaken by using elements having a fixed domain shape, whatever is the image content. By this change of reference, an interpolation is required between the pixels in each element (only on sub-images of whole picture) and the values assumed in the r - s coordinate system. According to the isoparametric formulation, the same interpolation used to locate any internal point is adopted to evaluate internal displacements. The use of r - s coordinate system allows also to manage non-regular shaped elements. And to modify the number of unknowns considered.

3.3 Comparison Between Grouping Technique and Hu's Invariant

In this section a comparison is proposed: (i) full pixel computation; (ii) grouping technique by varying packaging dimension; (iii) Hu's invariant moments. All techniques have been applied on the same reference image. This image shows the surface of a granite (Fig. 13). The use of this image is due to his particular distribution of color that is a natural sort of speckle. The image is divided in 3×3 elements and 16 nodes; the dimension of a single element is 100×100 pixels. The original image is digitally deformed by $\varepsilon_x = 0.02$ and $\varepsilon_y = 0.01$. The (i) results (no enhancement) are obtained when the number of packets (a packet contains adjacent pixel grouped together) is equal to 10000 (number of pixels in each sub-image).

Several tests are performed progressively decreasing the number of packets. The lowest number of packets considered is 9, while the maximum numbers is 10000, representing the solution without any enhancement technique. Both convergence time and final error of the displacements are compared. All computations are performed on an internal processor Intel® Core i7-2600K having 3.4 GHz.

The error is calculated through a ratio: the numerator is the sum of the displacement differences between the identified final nodal position and the theoretical one (known due to strain imposed); the denominator is the sum of the differences between the theoretical position and the initial one.

In Fig. 14 errors and convergence times are normalized by respect to the highest values encountered. Note that this relative definition of error penalizes the lower displacements and this must be kept in mind when comparing different deformation magnitudes.

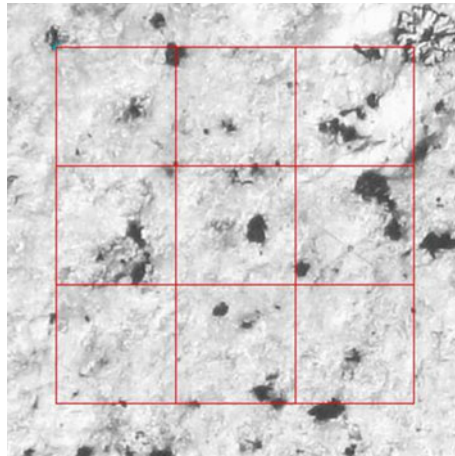


Fig. 13 Reference undeformed image 500 × 500 pixel

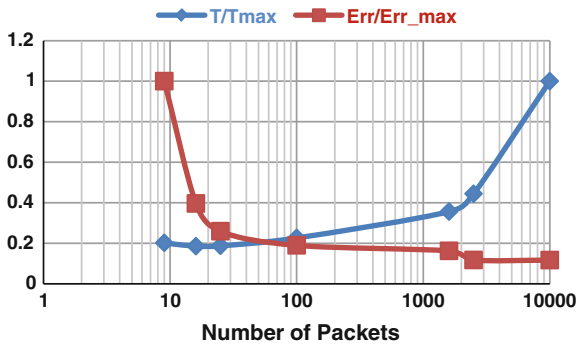


Fig. 14 Cpu Time versus error for the grouping technique for the grouping technique

As expected, the maximum errors are obtained with the minimum number of packets; the worst convergence time is obtained when no enhanced technique is used. Increasing the number of packets decreases the error convergence, but increases the calculation time and vice versa. It is possible to detect a crossing point of minimum error-time curves (Fig. 14, Table 5). It is interesting to highlight that the red curve (squares) identifies a well-defined knee, demonstrating that a strong grouping is possible while shortly affecting accuracy.

The results obtained with the method of Hu’s invariant moments are shown in Table 6. This method does not show at the present time particularly good outcomes. The invariants allow to greatly reducing the number of equations of the system, but they do not ensure acceptable results both in terms of computing time and precision of the solution. Even the use of non-speckle images helped to gain accurate results.

Table 5 Grouping technique performance

Number of packs	Iterations	Total times[s]	Jacobian Time[s]	Error[%]	Average Error [pixel]
4			Not convergence		
9	12	0,8886	0,0243	8,1729	0,2311
16	11	0,8166	0,0242	3,7483	0,1060
25	10	0,8252	0,0242	2,6902	0,0760
100	12	0,9963	0,0256	1,5607	0,0441
1600	13	1,5676	0,0531	1,4710	0,0416
2500	13	1,9551	0,0728	1,0258	0,0290
No enhancement	17	4,4014	0,2526	1,0848	0,0306

Table 6 Hu's invariant moments technique performance

Number invariants utilized	Iterations	Total time[s]	Jacobian time[s]	Error[%]	Average error[pixel]
14	236	20,7042	0,0367	10,0316	0,2837
12	236	20,7139	0,0382	10,0397	0,2840
10	258	22,6446	0,0384	22,3302	0,6316

Table 7 Grouping technique performance

Number of packs	Iterations	Total time[s]	Jacobian time[s]	Error[%]	Average error[pixel]
25	26	39,7946	0,1358	7,2077	0,4189
No enhancement	410	926,9013	10,2175	6,7912	0,3947

However, at strain values of the order of some percent, the Hu's invariant method works acceptably. The increase of the convergence time is mainly due to the increment in the number of iterations required to converge (Table 7).

Further tests have been performed using the grouping technique. In particular, some tests considered large strains applied, (over 30%). It is interesting to observe that the grouping technique is able to manage also this amount of image differences, even when the non-enhanced technique is unable to reach convergence. Thus, the use of packets helps to speed up the solution time, but it is also skillful to organize information so that the convergence capability is stronger than before. As an example, in the case analyzed and proposed in Fig. 13, presenting a strong deformation reaching 0.35 in both principal directions, calculated with 25 packages, the method returns a solution with an error close to 0.56 % and 9.84 s of convergence time, whereas when non-using packages the solution does not converge at all (Fig. 15).

To validate the grouping technique, another example is performed: the image used is not a speckle, but a generic image (canvas paint in Fig. 16). In this case the deformations are not of the same magnitude of the preceding ones, but they are set to $\varepsilon_x = 0.03$ and $\varepsilon_y = 0.01$; much more elements are used to mesh the image.

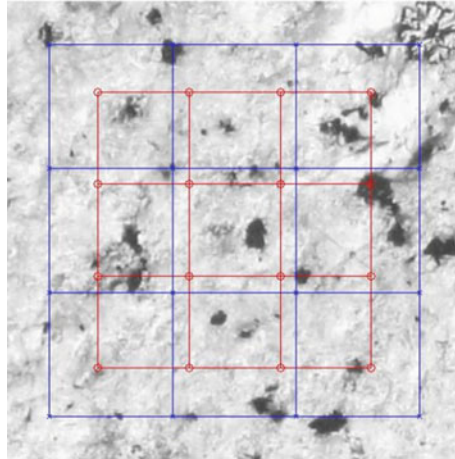


Fig. 15 Example large deformed image and grid solution obtained by grouping technique

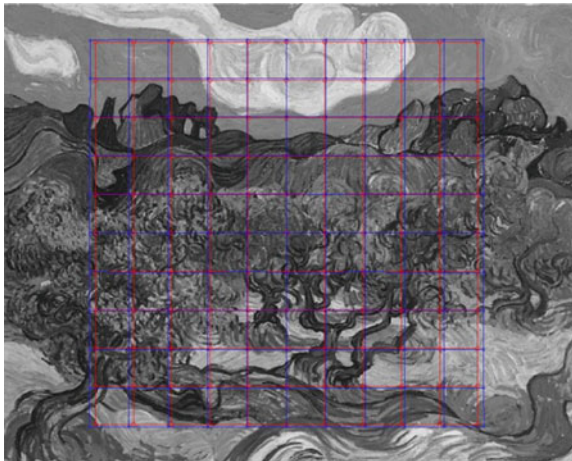


Fig. 16 Example deformed image and grid 10×10 solution obtained by grouping technique

In particular a grid of 10×10 elements, having 50×50 pixels each is used. The study was performed both with a number of packets equal to 25, and without any convergence enhancement technique.

This latter example shows once again the convenience of the use of packets, both in terms of accuracy and computational time. This convenience is stronger and stronger when increasing the number of sub-images managed. Incidentally, it is shown that the use of packets is profitable on non-speckled images.

4 Conclusions

In this work it has been discussed a discretization technique that is able to account of off-plane displacements of a surface as well as in-plane deformations. For both circumstances a differential method to determine the displacement field is presented. In the chapter two different procedures to reduce the number of equations are discussed. The first method consists on the grouping of the Jacobian matrix. Despite the reduction of the number of equations, the system is always over-determined, the solution converges with an error decreasing while increasing the number of packets. This method is robust and reliable and allows convergence even when very high deformations occur.

A possible use is to apply this procedure as a preliminary calculation in order to approach the exact solution, then refining the results by omitting packets grouping. The second method uses Hu's invariants for the assembly of the Jacobian matrix. The number of the equations is reduces to the number of Hu's invariant moments. This second method designed to enhance the convergence is not as reliable as the previous one; as a matter of fact, even though the number of equations is lower than in the grouping method, the iterations required increase significantly. Furthermore, the increase in the total time is not accompanied by accuracy growth.

References

1. Amodio D, Broggiato GB, Salvini P (1995) Finite strain analysis by image processing: smoothing techniques. *Strain* 31(3):151–157
2. Broggiato GB, Cortese L, Santuci G (2006) Misura delle deformazioni a campo intero attraverso l'elaborazione di sequenze di immagini ad alta risoluzione. In: XXXV Convegno Nazionale AIAS
3. Cheng P, Sutton MA, Schreier HW, McNeill SR (2002) Full-field speckle pattern image correlation with B-spline deformation function. *Exp Mech* 42(3):344–352
4. Cofaru C, Philips W, Van Paepegem W (2010) Improved Newton-Raphson digital image correlation method for full-field displacement and strain calculation. *Appl Opt* 49:33
5. Dougherty G (2009) *Digital image processing with medical applications*. Cambridge University Press, Oxford
6. Flusser J, Suk T, Zitová B (2009) *Moments and moment invariants in pattern recognition*. Wiley, New York
7. Hild F, Roux S (2008) A Software for "Finite-element" displacement field measurements by digital image correlation. Internal report n. 269, LMT-Cachan, UniverSud Paris
8. Hu MK (1962) Visual pattern recognition by moment invariants. *IRE Trans Info Theory* IT-8:179–187
9. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. *Int J Comput Vis* 60(2):91–110
10. Lu H, Zhang X, Knauss WG (1997) Uniaxial, shear, and poisson relaxation and their conversion to bulk relaxation: studies on poly(methyl methacrylate). *Polym Compos* 18(2):211–222
11. Pilet J, Lepetit V (2007) Fast non-rigid surface detection. *Registration and realistic augmentation*. *Int J Comput Vis* 76(2):109–122
12. Lagattu F, Brillaud J, Lafarie-Frenot M (2004) High strain gradient measurements by using digital image correlation technique. *Mater Charact* 53(2004):17–28

13. Sutton MA, Cheng MQ, Peters WH, Chao YJ, McNeill SR (1986) Application of an optimized digital correlation method to planar deformation analysis. *Image Vis Comput* 4(3):143–150
14. Sutton MA, McNeill SR, Helm ID, Schreier HVr, Chao YJ (2000) Photomechanics. In: Rastogi PK (ed) *Advances in 2D and 3D computer vision for shape and deformation measurements*. Topics in applied physics, vol 77. Springer, New York, pp 323–372
15. Sutton MA, Ortu JJ, Shreier HW (2009) *Image correlation for shape. Motion and deformation measurements, basic concept, theory and applications*. Springer, New York
16. Zienkiewicz OC, Taylor RL (1967) *The finite element method*, McGraw-Hill, New York