

# Dense and accurate motion and strain estimation in high resolution speckle images using an image-adaptive approach

Corneliu Cofaru<sup>a</sup>, Wilfried Philips<sup>a</sup> and Wim Van Paepegem<sup>b</sup>

<sup>a</sup>Ghent University, Telin - IPI - IBBT, St-Pietersnieuwstraat 41, B-9000 Gent, Belgium;

<sup>b</sup>Ghent University, Department of Materials Science and Engineering, St-Pietersnieuwstraat 41, B-9000 Gent, Belgium;

## ABSTRACT

Digital image processing methods represent a viable and well acknowledged alternative to strain gauges and interferometric techniques for determining full-field displacements and strains in materials under stress. This paper presents an image adaptive technique for dense motion and strain estimation using high-resolution speckle images that show the analyzed material in its original and deformed states. The algorithm starts by dividing the speckle image showing the original state into irregular cells taking into consideration both spatial and gradient image information present. Subsequently the Newton-Raphson digital image correlation technique is applied to calculate the corresponding motion for each cell. Adaptive spatial regularization in the form of the Geman-McClure robust spatial estimator is employed to increase the spatial consistency of the motion components of a cell with respect to the components of neighbouring cells. To obtain the final strain information, local least-squares fitting using a linear displacement model is performed on the horizontal and vertical displacement fields. To evaluate the presented image partitioning and strain estimation techniques two numerical and two real experiments are employed. The numerical experiments simulate the deformation of a specimen with constant strain across the surface as well as small rigid-body rotations present while real experiments consist specimens that undergo uniaxial stress. The results indicate very good accuracy of the recovered strains as well as better rotation insensitivity compared to classical techniques.

**Keywords:** Digital Image Correlation (DIC), speckle image, adaptive cells, Newton-Raphson, regularization

## 1. INTRODUCTION

Since their introduction in the early 80's,<sup>1-3</sup> Digital image correlation (DIC) methods have become a reliable measurement tool for the full-field measurement of displacements and strains in the field of experimental mechanics. Various algorithmic approaches have been developed over time, the most prominent of these involving the Newton-Raphson iterative minimization technique.<sup>4-6</sup> Respect to the other existing methods, it presents better sub-pixel accuracy and the possibility of being combined with more complex linear and quadratic motion models at the expense of computational speed and sensitivity to the interpolation employed.<sup>7-9</sup> However, these disadvantages are becoming less problematic in present experiments because of the continuous evolution of computing hardware performances and the fact that interpolants such as high order splines<sup>10,11</sup> significantly reduce the negative impact of interpolation in the motion and strain estimates. In order to obtain the displacements, two images of the analyzed material specimen - which is spray-painted to present a seemingly random speckle pattern, taken before (*reference image*) and after (*deformed image*) the deformation process are used. The reference image is divided into rectangular blocks and motion is calculated by finding the positions of the reference blocks into the deformed image. The strains and subsequently obtained from the displacement fields through differentiation of the latter. A determining factor in the final accuracy of the DIC results is the size of the blocks into which the image is partitioned. If large blocks are used, slow spatial variations in the underlying motion fields are accurately captured however, faster ones are smoothed. Smaller blocks can capture fast spatial

---

Further author information: (Send correspondence to C. C.)

C.C.: E-mail: Cornel.Cofaru@telin.ugent.be, Telephone: +32 9 264 3416

W.P.: E-mail: Wilfried.Philips@telin.ugent.be, Telephone: +32 9 264 3385

W.V.P.: E-mail: Wim.VanPaepegem@UGent.be, Telephone: +32 9 264 4207

variations but accuracy is lowered by effects such as the aperture problem and the fact less image information is used.

This paper presents a new alternative to the block-based DIC approach which aims at improving overall accuracy while maintaining the effectiveness in capturing fast spatial motion fluctuations. The new Newton-Raphson -based DIC approach employs spatially irregular image *cells* adaptively built according to the speckle pattern present in the image to calculate motion. Moreover, spatial regularization is introduced to increase the spatial consistency of the displacement components corresponding to neighbouring cells. Compared to the original block-based method, the use of irregular cells and regularization allows for a more efficient use of the image information improving overall accuracy. The theoretical aspects of the new method are presented in Section 2 with the experimental results and conclusions in Sections 3 and 4 respectively.

## 2. THE CELL-BASED NEWTON-RAPHSON METHOD

### 2.1 Image partitioning

The aim of the image partitioning is to divide the image into irregular cells so that each cell contains inside at least a speckle large enough to allow accurate motion recovery. This approach is especially suited for hi-resolution and hi-magnification images where the number of speckles whose sizes satisfy the above condition is considerable. The partitioning algorithm can be informally split into two phases: the first detects large speckles or agglomerations of speckles with a diameter over a certain size  $d_{\min}$ . The speckle *central areas* - here considered the results of a morphological thinning operation - are each distinctly labeled. The second phase assigns each of the remaining image pixels one of the labels computed previously which effectively partitions the image into cells. The label assignment operation minimizes a distance that takes into consideration both spatial distances and the image gradient of the pixels found on the path between the pixel to be labeled and the closest pixels of distinct previously labeled speckle central areas. The end result is a label map  $L(x, y)$  the same size as the reference image. The value at each spatial location  $(x, y)$  indicates to which cell the pixel in the reference image found at that location belongs to.

First, a binary image is created by applying Otsu's method<sup>12</sup> to the reference image. The binary image is eroded with a circular element  $d_{\min}$  pixels in size to control the density and size of the cells and morphological thinning is applied. The last step in the first phase of the partitioning algorithm is a connected component analysis using an 8-pixel connected neighbourhood. It assigns the label, which is a number placed in the label map, to all pixels belonging to the same "thinned" binary speckle. In the second partitioning phase, a pixel to be partitioned located at  $(x, y)$ ,  $N_C$  components  $\{\mathbf{C}_i | i = 1 \dots N_C, 1 < N_C \leq N_L\}$  closest spatially to the pixel with  $N_L$  the total number of components in the image and  $\{(x_{\mathbf{C}_i}, y_{\mathbf{C}_i}) | i = 1 \dots N_C, 1 < N_C \leq N_L\}$  the locations in each connected component closest to the pixel at  $(x, y)$  are considered. The label map at  $(x, y)$  is assigned the value one of the connected component which minimizes:

$$D_i(x, y) = (1 - \lambda(x, y))E_i(x, y)S_i(x, y) + \lambda(x, y)S_i(x, y) \quad (1)$$

where:

$$E_i(x, y) = \frac{\sqrt{(x - x_{\mathbf{C}_i})^2 + (y - y_{\mathbf{C}_i})^2}}{\max_i \left( \sqrt{(x - x_{\mathbf{C}_i})^2 + (y - y_{\mathbf{C}_i})^2} \right)} \quad (2)$$

is the normalized Euclidean distance between the pixel at  $(x, y)$  and the location  $(x_{\mathbf{C}_i}, y_{\mathbf{C}_i})$  of the component  $\mathbf{C}_i$ . The term  $S_i$  with

$$S_i(x, y) = \sum_{j=1}^{N_{\theta_i}} \frac{\partial I}{\partial \theta_i} (x_{i,j}, y_{i,j})^2 \quad (3)$$

where  $N_{\theta_i} = \max(|x - x_{\mathbf{C}_i}|, |y - y_{\mathbf{C}_i}|)$  is the total number of pixels  $I(x_{i,j}, y_{i,j})$  in the image on the direct path of direction  $\theta_i$  that connects the locations, reflects the magnitude of the image gradient information found between  $(x, y)$  and  $(x_{\mathbf{C}_i}, y_{\mathbf{C}_i})$ . A graphical representation of the calculus of  $D_i$  is shown in Figure 1. The first term in  $D_i$  ensures that pixels located inside the speckles will always be in the same cell since the distance is kept very small due to the lack of gradient inside the speckle. The presence of image gradients at the speckle borders and larger

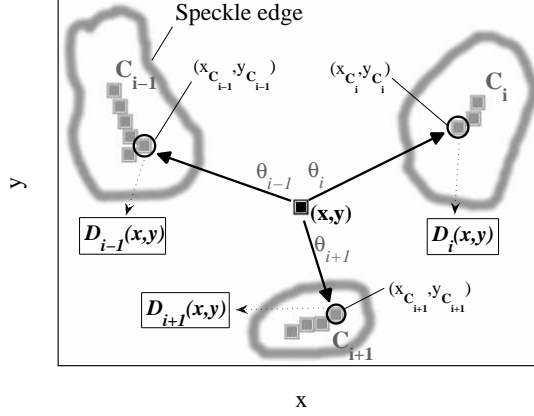


Figure 1: Graphical representation of the calculus of  $D_i$ .

Euclidean distance between the pixel and the speckle's central area increase  $D_i$  resulting in the final cell being too small fully contain the speckle edges. The term  $\lambda(x, y)$  compensates for this effect by lowering the influence of the spatial distance. Here,

$$\lambda(x, y) = \begin{cases} 0 & \text{if } \beta(x, y) \leq \alpha(x, y) \\ (\beta(x, y) - \alpha(x, y))^2 & \text{otherwise} \end{cases} \quad (4)$$

where  $\alpha(x, y)$  and  $\beta(x, y)$  are the two eigenvalues of the structure tensor of a  $M \times M$  pixel block centered in  $(x, y)$ , each normalized with respect to its highest value for the whole image. Further algorithmic details and analysis of the partitioning algorithm can be found in previous works.<sup>13</sup> In Figure 2 the results of each of the partitioning steps are presented using a  $300 \times 300$  pixel image. A binary mask that specifies if a pixel is to be partitioned or not can be also provided to the partitioning algorithm. Through this method, complex shapes can be partitioned provided that there is a straight path between the pixel and a connected component in the image that falls completely within the partitionable area.

## 2.2 Motion and strain estimation

The classical block-based Newton-Raphson method is extended here to use the pixels found within image cells when calculating the displacements. Also, robust displacement field regularization<sup>14-16</sup> is used to increase the spatial consistency of motion estimates. Its recent DIC employment<sup>17</sup> in the form of a robust function applied to the local differences between motion components showed great potential in improving displacement and accuracy while maintaining fast spatial variations intact. For the description of the motion estimation algorithm, a partitioned reference and a deformed speckle image  $F(x, y)$  and  $G(x', y')$  along with the  $i$ -th reference cell in the reference image,  $f_i(x, y) = \{F(x, y) | L(x, y) = i\}$  containing  $n_i$  pixels are considered. The integer pixel displacement of the center of mass of the reference image cell is initially calculated using a block-based approach in which the cross-correlation coefficient is maximized. If  $(x_0, y_0)$  is the position of the center of mass with  $x_0 = \frac{1}{n_i} \sum_{j=1}^{n_i} x_j$  and  $y_0 = \frac{1}{n_i} \sum_{j=1}^{n_i} y_j$  and  $u_0(x_0, y_0)$  and  $v_0(x_0, y_0)$  are the horizontal and vertical integer pixel displacements, a deformed cell  $g_i(x', y')$  with the same spatial shape is created inside the deformed image around the position  $(x'_0, y'_0)$  with  $x'_0 = x_0 + u_0(x_0, y_0)$ ,  $y'_0 = y_0 + v_0(x_0, y_0)$  as shown in Figure 3. Considering a linear displacement model where  $x' = x + u(x, y)$ ,  $y' = y + v(x, y)$  and

$$u(x, y) = P_1 + P_3(x - x_0) + P_5(y - y_0) \quad (5)$$

$$v(x, y) = P_2 + P_4(x - x_0) + P_6(y - y_0) \quad (6)$$

are the sub-pixel horizontal and vertical displacements of the reference cell's pixels respect to their corresponding deformed cell positions, the error functional between the cells can be defined as:

$$E(\mathbf{P}) = E_D(\mathbf{P}) + \lambda E_S(\mathbf{P}) \quad (7)$$

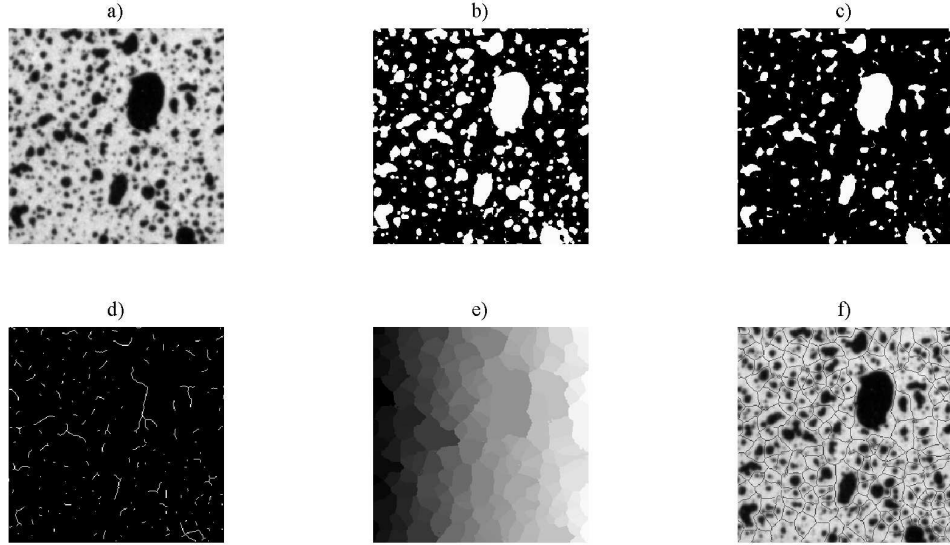


Figure 2: a) Original image  $300 \times 300$  pixels in size; b) Binary image; c) Results after erosion with a circular element with  $d_{\min} = 2$  pixels; d) Morphological thinning results; e) Final label map  $L(x, y)$ ; f) Position of cell borders (dark grey) in the image. In the test,  $N_C = 2$  and  $M = 5$  were used.

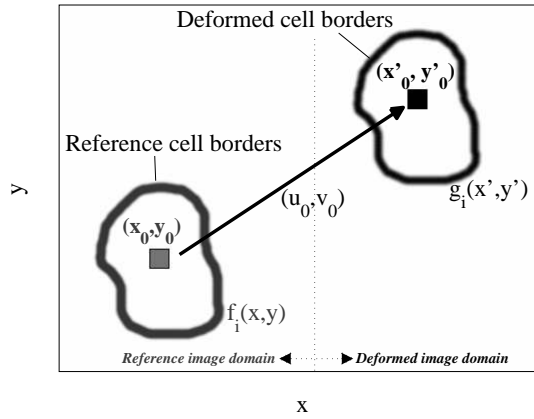


Figure 3: The construction of a *deformed cell*.

with  $\mathbf{P} = \{P_1, \dots, P_6\}$  the six displacement components. The term

$$E_D(\mathbf{P}) = \sum_{(x,y) \in f_i} (f_i(x, y) - g_i(x, y, \mathbf{P}))^2 \quad (8)$$

is a least-squares measure of the image data fit between the two cells,  $\lambda$  is the regularization strength parameter while

$$E_S(\mathbf{P}) = \sum_{j=1}^6 \sum_{k=1}^{N_{f_i}} \rho(r_{jk}, \sigma_j(r_{jk})) \quad (9)$$

is the regularization term. The Geman-McClure robust function  $\rho(r, \sigma) = r^2 / (r^2 + \sigma)$  controls how much each of the six motion components  $P_j$  associated with a cell is influenced by the corresponding components of the  $N_{f_i}$

neighbouring cells. The parameter  $\sigma_j$  which is a function of the standard deviation  $\tilde{\sigma}_j$  of the local smoothness residuals  $r_{jk} = \{P_j - P_j^{(k)} | j = 1 \dots 6, k = 1 \dots N_{f_i}\}$  adjusts this influence by changing the shape of the robust function. A linear dependency  $\sigma_j = 15 \cdot \tilde{\sigma}_j$  has been chosen throughout the tests. The choice of the multiplicative factor represents a compromise between the discontinuity preservation property of the regularization term which is more pronounced for smaller values and the smoothing effect which is stronger for larger values. Minimizing Equation 7 is done iteratively, with the solution at the  $t$ -th iteration of the form:

$$\mathbf{P}^t - \mathbf{P}^{t-1} = -\frac{\nabla E(\mathbf{P}^{t-1})}{\nabla \nabla E(\mathbf{P}^{t-1})} \quad (10)$$

where

$$\nabla E(\mathbf{P}) = \left[ \left( \frac{\partial E_D}{\partial P_1} + \lambda \frac{\partial E_S}{\partial P_1} \right) \dots \left( \frac{\partial E_D}{\partial P_6} + \lambda \frac{\partial E_S}{\partial P_6} \right) \right] \quad (11)$$

and

$$\nabla \nabla E(\mathbf{P}) = \begin{bmatrix} \left( \frac{\partial^2 E_D}{\partial P_1^2} + \lambda \frac{\partial^2 E_S}{\partial P_1^2} \right) & \dots & \left( \frac{\partial^2 E_D}{\partial P_1 \partial P_6} \right) \\ \vdots & \ddots & \vdots \\ \left( \frac{\partial^2 E_D}{\partial P_6 \partial P_1} \right) & \dots & \left( \frac{\partial^2 E_D}{\partial P_6^2} + \lambda \frac{\partial^2 E_S}{\partial P_6^2} \right) \end{bmatrix} \quad (12)$$

are the Jacobian and Hessian matrices of the error functional  $E(\mathbf{P})$ . Practically, in each iteration the reference cell is warped towards the deformed cell and the residual displacement components between the two are calculated and added to the previous iteration solution. Convergence is considered to be reached when the difference between all corresponding motion components in two successive iterations is smaller than  $10^{-5}$ . To deform the reference cell, bicubic spline interpolation is used. The final horizontal and vertical displacement fields obtained have the same resolution as the reference image. To obtain strain fields that can be directly compared with the classic block-based method, the displacement fields are subsampled at the locations that correspond to the centers of the blocks. In this way, the displacement fields from both the proposed and classic block-based methods will have the same size and the displacements will correspond to the same locations in the reference image. The strain fields can be calculated in a straightforward manner<sup>18</sup> by least-squares fitting linear motion models through the horizontal and vertical displacements located in a rectangular window also called *strain window*. After the fit has been performed, the horizontal, shear and vertical strain values corresponding to the strain window can be easily calculated function of the linear model parameters.

### 3. TESTS AND RESULTS

#### 3.1 Numerical experiments

The first numerical experiment simulates constant horizontal and vertical strains  $\varepsilon_{xx} = \varepsilon_{yy} = 5 \times 10^{-3}$  throughout the image while the second experiment simulates a vertical strain  $\varepsilon_{yy} = 7 \times 10^{-3}$  and a clockwise rotation of  $0.5^\circ$  around the upper left corner of the image. The deformed image in these cases is obtained by warping a  $512 \times 512$  pixel reference image using a radial basis function interpolation realistic deformation framework.<sup>9</sup> The reference image is part of a larger  $2208 \times 3000$  pixel frame captured with a Pixelink PL-A782 camera in a previous experiment. Through the optical alignment, one image pixel corresponds to  $8.33 \mu\text{m}$  in the object plane. The reference image and contours of the cells are shown in Figure 4. An erosion element with  $d_{\min} = 1$  pixels in size,  $N_C = 3$  and  $M = 5$  were used resulting in 573 cells. A larger diameter of the erosion element would have generated larger cells however the average size of the resulting cells, equivalent to a  $21.3 \times 21.3$  pixel block, was large enough to obtain reliable motion estimates. For an objective comparison with block approaches, both cells and blocks will have similar sizes. In consequence, overlapping blocks of  $21 \times 21$ ,  $23 \times 23$  and  $25 \times 25$  pixels with a 7 pixel step size will be employed by the block-based method. Strains are calculated using five different strain windows with sizes between  $7 \times 7$  and  $15 \times 15$  displacements. Strain accuracy is assessed by calculating the mean of the absolute strain errors for the block and strain window sizes used in each of the numerical experiments. This provides a good quantitative evaluation on the overall quality of the estimated deformation and can reliably indicate accuracy differences if present. The regularization parameter for the cell-based DIC method had the

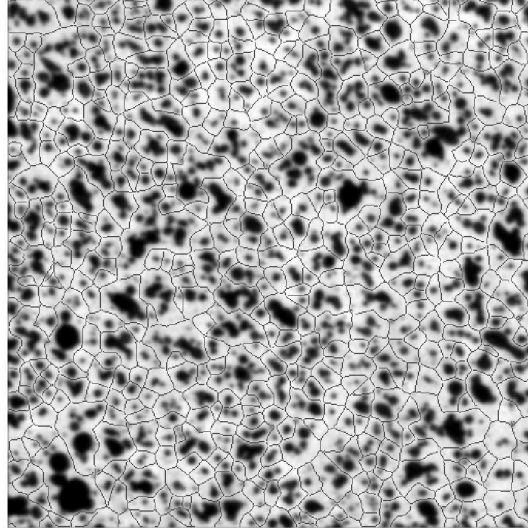


Figure 4: The reference image used in the first two (numerical) experiments with the contours of the resulting cells.

values of  $\lambda = 0$  and  $\lambda = 100$  to separately assess the influence of the adaptive cells and regularization. All the methods presented were implemented in Matlab on a Core2 Duo 2.0 GHz machine running the Linux operating system. As can be seen from the mean absolute errors from Figure 5, the proposed method yields the best accuracy regardless of the strain window size used. It is interesting to note that the method performs better than the block-based one with  $25 \times 25$  pixel blocks which are larger than the average cell size suggesting that on average, the cells contain more image information than the blocks. Furthermore, the fact that the cells do not overlap does not disadvantage the method. As expected, the strain errors decrease as the strain window sizes increases. This is because the more displacement values are used, the more accurate the linear fitting parameters will be. The mean vertical and horizontal strain errors for this case present similar magnitudes for the same strain window sizes, with errors generally varying between approximately 3% of the  $5 \times 10^{-3}$  strain amplitudes for the  $7 \times 7$  strain windows and 0.7% for the  $13 \times 13$  strain windows. It is important to note that in real experiments these errors are generally larger due to noise, lighting and quantization effects associated with the image acquisition process. The results of the second experiment, shown in Figure 6 are consistent with the ones from the first although certain differences exist: in the case of the horizontal strain errors, strain windows larger than  $11 \times 11$  do not bring any qualitative improvements with the proposed method still yielding the lowest overall error. The proposed method performs considerably better for the shear strains with parasitic shear strains - which are due to the introduced rotation - up to half the size of the ones generated by the block-based method in the case of the  $9 \times 9$  strain windows. Vertical strain errors were slightly larger than the ones of the block-based method with  $25 \times 25$  pixel blocks however the minimum error was still achieved by the proposed method using  $15 \times 15$  strain windows.

### 3.2 Validation in real experiments

The experimental validation is done by using the proposed DIC method to measure full-field strains in two experiments. Both consist in the application of uniaxial load in a vertical upward direction on aluminum specimens using a Instron 8801 servo-hydraulic machine. The specimens are spray-painted to obtain a black speckle pattern, mounted in the servo-hydraulic machine and photographed using the Pixelink PL-A782 camera at a resolution of  $2208 \times 3000$  pixels before and after deformation. The specimen used in the third experiment is rectangular with two lateral slits as seen in Figure 7a which shows the reference image sized  $1000 \times 1000$  pixels used in the measurement process. A detail sized  $500 \times 500$  pixels of the partitioning results is shown in Figure 7b. The

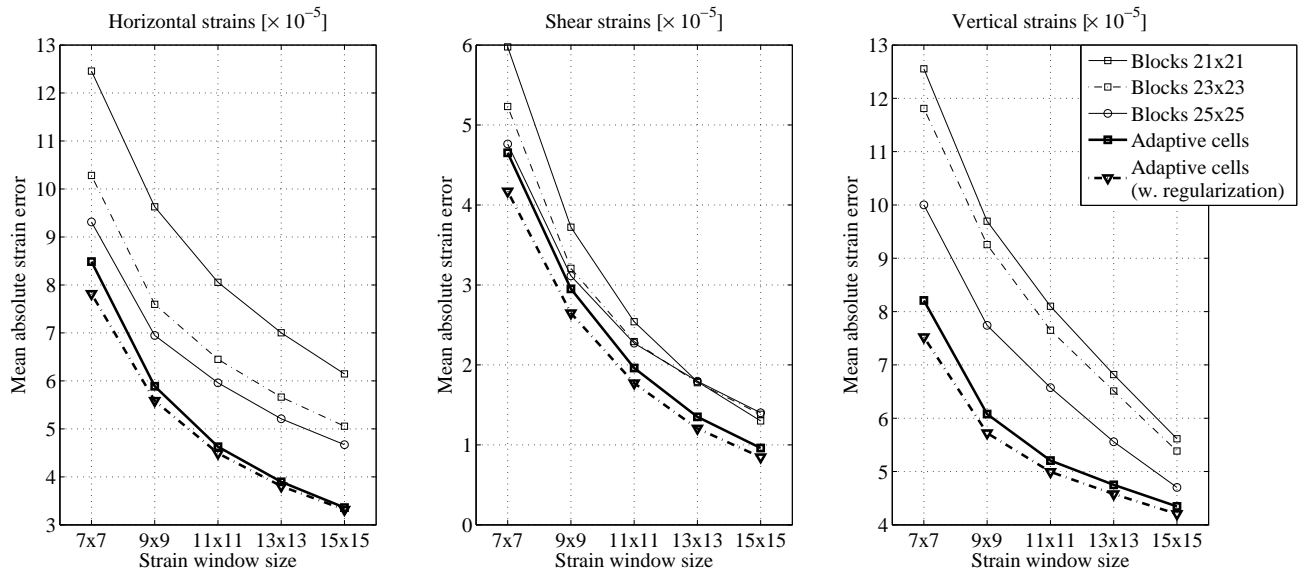


Figure 5: Mean absolute horizontal, shear and vertical strain errors for different strain windows sizes in the first (numerical) experiment.

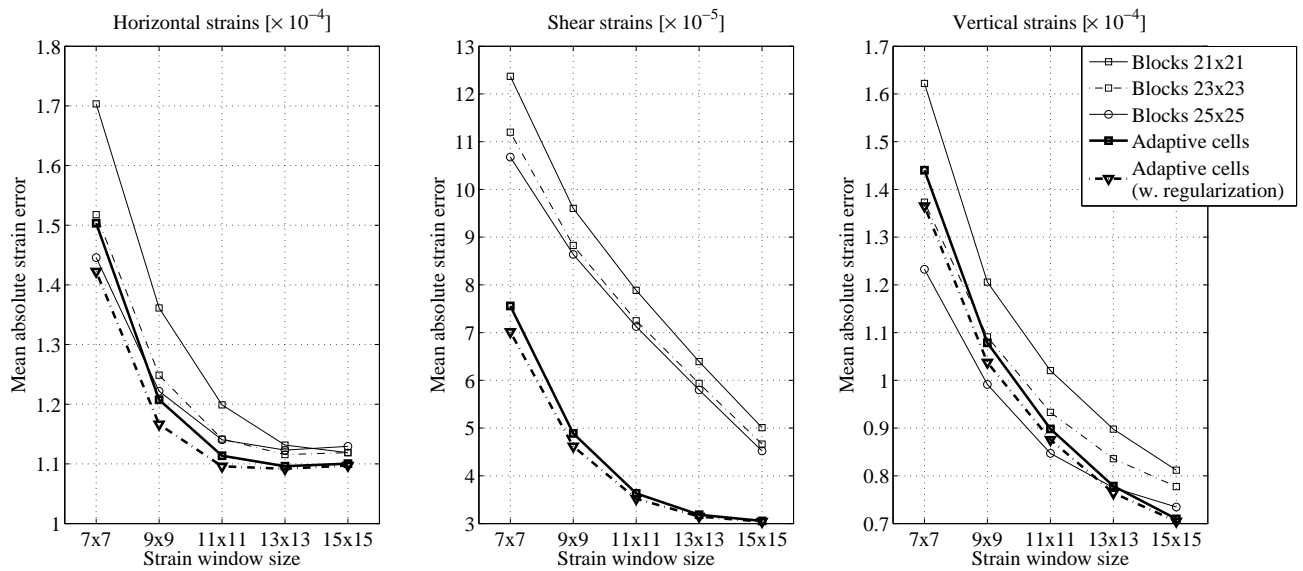
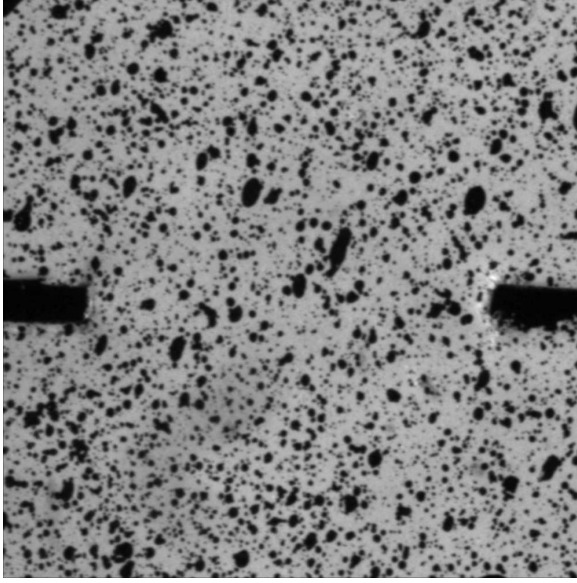
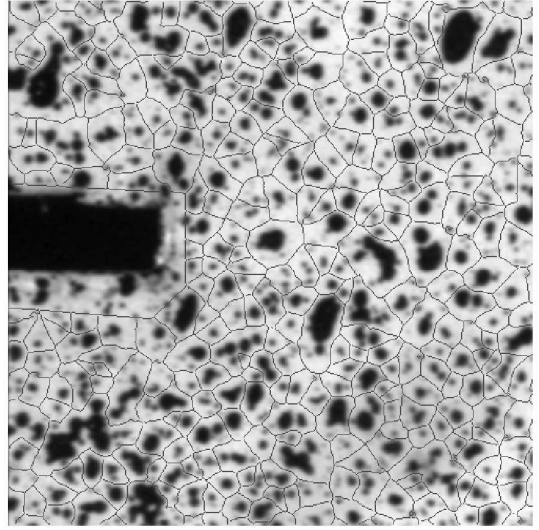


Figure 6: Mean absolute horizontal, shear and vertical strain errors for different strain windows sizes in the second (numerical) experiment.



(a) Reference image for the third experiment(1000 × 1000 pixels)



(b) Partitioning detail (500 × 500 pixels)

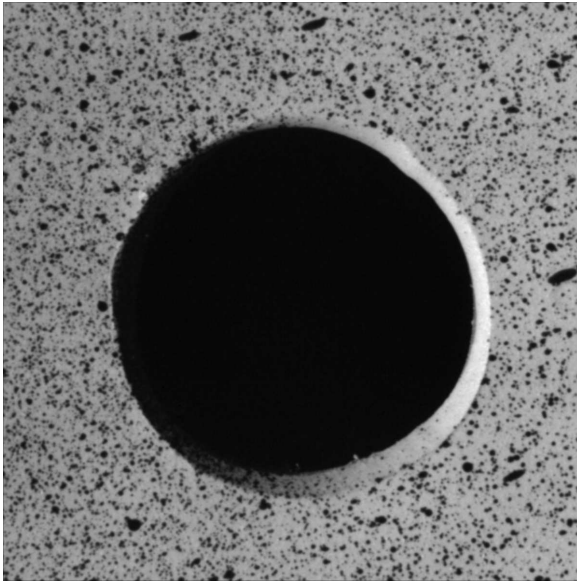
Figure 7: Third experiment reference image and partitioning detail.

fourth experiment used an aluminum specimen with a central hole. The reference image sized  $1200 \times 1200$  pixels and a detail of the partitioning results sized  $500 \times 500$  pixels can be seen in Figures 8a and 8b respectively. The partitioning and motion estimation algorithms used the same sets of parameters in both validation experiments:  $d_{\min} = 2$  pixels,  $N_C = 3$  and  $M = 5$ ,  $\lambda = 100$  and a strain window size of  $13 \times 13$  displacements. The number of cells resulted from the image partitioning was 745 for the third and 881 for the fourth experiment. The horizontal and vertical strain fields from the two experiments shown in Figures 9 and 10 respectively, indicate as expected the largest strain concentrations in the central areas of the specimens. For the third experiment, measured strain amplitudes vary between  $-3.2\%$  and  $2.3\%$  for the horizontal strains and from  $3.5\%$  in-between the slits to  $8\%$  near the tip of the slits for the vertical strains. The large positive vertical strain values indicate that the specimen deformed only in the area in-between the lateral slits, the load applied elongating the specimen. In the fourth experiment strains are present in the area adjacent to the central hole's left and right extremities. The vertical strains are positive with maximum values up to  $4.5\%$  indicating stretching. The horizontal strains are negative with values down to  $-1.7\%$  indicating compression. This is consistent with the fact that as the specimen is pulled upwards, it's middle section moves inwards, compressing the material. At this stage, there is no way of assessing the accuracy of the method because no ground truth data exists however, the results are in line with the expected material behaviour under the specific loads applied.

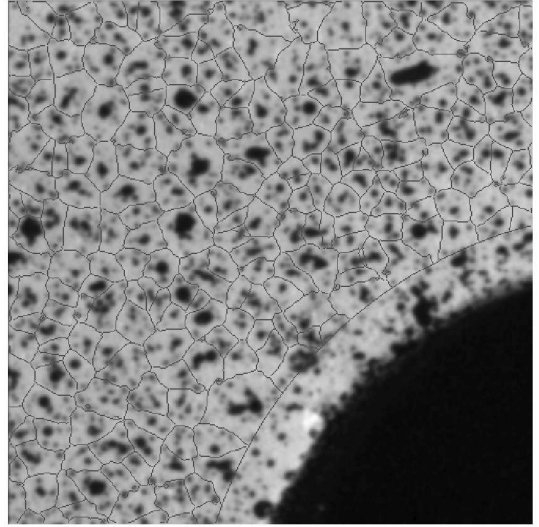
#### 4. CONCLUSION

In this paper, the concepts of irregular image cells and robust regularization have been combined and introduced in the Newton-Raphson DIC method with the aim of improving motion and strain accuracy. The newly resulted DIC method was evaluated in numerical experimental simulations in which the displacements and strains were known and validated in real experimental scenarios. Results indicate that the improvements in accuracy are indeed possible if the rectangular blocks are replaced by irregular shaped cells, the new method providing higher accuracy than the conventional approach. The inclusion of the regularization term further improved the accuracy.



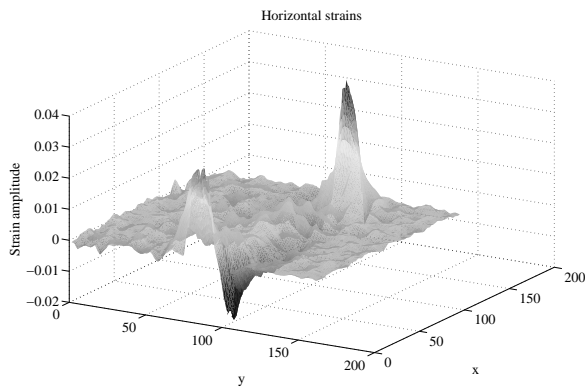


(a) Reference image for the fourth experiment ( $1200 \times 1200$  pixels)

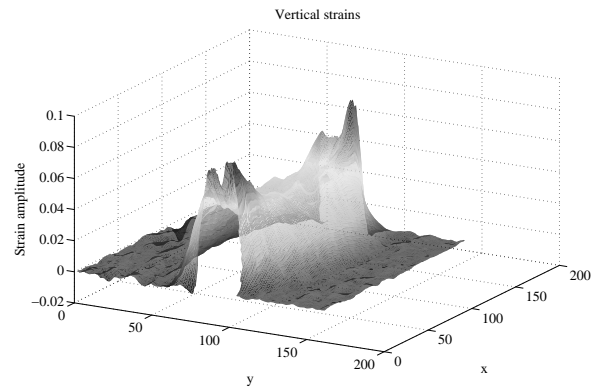


(b) Partitioning detail ( $500 \times 500$  pixels)

Figure 8: Fourth experiment reference image and partitioning detail.



(a) Horizontal strains



(b) Vertical strains

Figure 9: Third experiment horizontal and vertical strains.

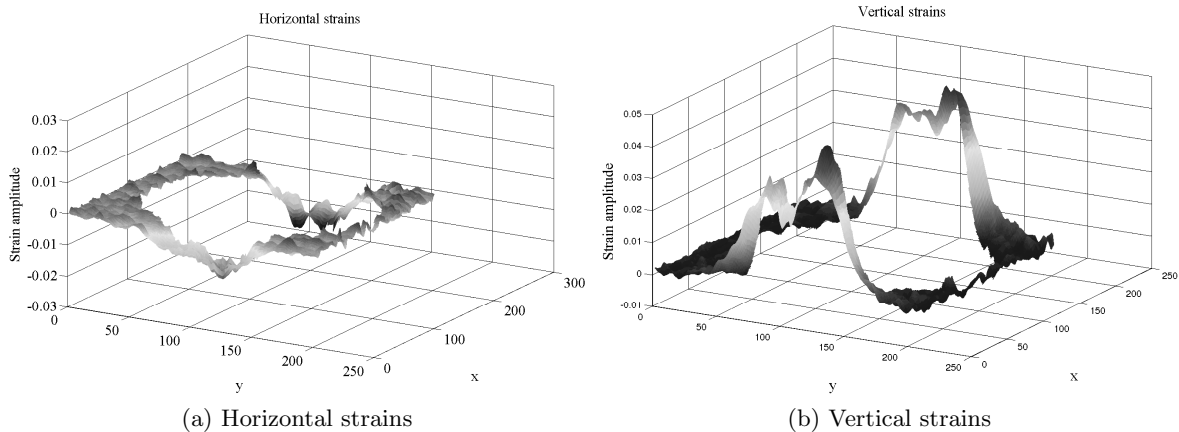


Figure 10: Fourth experiment horizontal and vertical strains.

In the experimental validation stage the method yielded results in accordance with theoretical assumptions about the real material behaviour under load.

## REFERENCES

- [1] Peters, W. and Ranson, W., “Digital imaging techniques in experimental stress analysis,” *Optical Engineering* **21**(3), 427–432 (1982).
- [2] Sutton, M., Wolters, W., Peters, W., and McNeill, S., “Determination of displacements using an improved digital correlation method,” *Image Vision Computing* **1**(3), 133–139 (1983).
- [3] Chu, T., Ranson, W., and M.A., S., “Applications of digital-image-correlation techniques to experimental mechanics,” *Experimental Mechanics* **25**(3), 232–244 (1985).
- [4] Bruck, H., McNeill, S., Sutton, M., and Peters III, W., “Digital image correlation using Newton-Raphson method of partial differential correction,” *Experimental Mechanics* **29**(3), 261–267 (1989).
- [5] Vendroux, G. and Knauss, W., “Submicron deformation field measurements: Part 2. improved digital image correlation,” *Experimental Mechanics* **38**(2), 86–92 (1998).
- [6] Lu, H. and Cary, P., “Deformation measurements by digital image correlation: Implementation of a second-order displacement gradient,” *Experimental Mechanics* **40**(4), 393–400 (2000).
- [7] Pan, B., Xie, H., Xu, B., and Dai, F., “Performance of sub-pixel registration algorithms in digital image correlation,” *Measurement Science & Technology* **17**(6), 1615–1621 (2006).
- [8] Pan, B., Qian, K., Xie, H., and Asundi, A., “Two-dimensional digital image correlation for in-plane displacement and strain measurement: a review,” *Measurement Science & Technology* **20**(062001 (17pp)) (2009).
- [9] Cofaru, C., Philips, W., and Van Paepegem, W., “Evaluation of digital image correlation techniques using realistic ground truth speckle images,” *Measurement Science & Technology* **21**(055102 (17pp)) (2010).
- [10] Sutton, M., Orteu, J.-J., and Schreier, H., [*Image Correlation for Shape, Motion and Deformation Measurements*], Springer (2009).
- [11] Reu, P., “Experimental and numerical methods for exact subpixel shifting,” *Experimental Mechanics* **51**, 443–452 (2011).
- [12] Otsu, N., “A threshold selection method from grey-level histograms,” *IEEE Trans. Syst., Man., Cybern* **9**, 62–66 (January 1979).
- [13] Cofaru, C., Philips, W., and Van Paepegem, W., “Adaptive partitioning method in high resolution speckle imagery for sub-pixel digital image correlation,” in [*Proceedings of the IEEE International Conference on Image Processing ICIP’10*], 769–772 (2010).

- [14] Black, M. and Anandan, P., “The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields,” *Computer Vision and Image Understanding* **63**(1), 75–104 (1996).
- [15] Ye, M., Haralick, R., and Shapiro, L., “Estimating piecewise-smooth optical flow with global matching and graduated optimization,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**(12), 1625–1630 (2003).
- [16] Teng, C.-H., Lai, S.-H., Chen, Y.-S., and Hsu, W.-H., “Accurate optical flow computation under non-uniform brightness variations,” *Computer Vision and Image Understanding* **97**(3), 315–346 (2005).
- [17] Cofaru, C., Philips, W., and Van Paepegem, W., “Improved Newton-Raphson digital image correlation method for full-field displacement and strain calculation,” *Applied Optics* **49**(33), 6472–6484 (2010).
- [18] Pan, B., Asundi, A., Xie, H., and Gao, J., “Digital image correlation using iterative least squares and pointwise least squares for displacement field and strain field measurements,” *Optics and Lasers in Engineering* **47**, 865–874 (2009).