A novel coarse-fine search scheme for digital image correlation method

Zhi-Feng Zhang a, Yi-Lan Kang a,*, Huai-Wen Wang a, Qing-Hua Qin a,b, Yu Qiu a, Xiao-Qi Li a

a Department of Mechanics, School of Mechanical Engineering, Tianjin University, Tianjin 300072, PR China
b Department of Engineering, Australian National University, Canberra, ACT 0200, Australia

Received 6 May 2004; received in revised form 15 March 2006; accepted 15 March 2006
Available online 28 March 2006

Abstract

As a well-established technique for measuring surface deformation, the digital image correlation method (DICM) has received wide application in experimental mechanics and structural analysis in recent years. One of the major challenges in practical applications of this technique is how to achieve the lowest probability of mis-identification and reduce computation cost simultaneously. Based on this understanding, a novel coarse-fine search method for correlation analysis is developed in this paper. This method is based on an affine transform and a new technique of fine searching called “nested fine search method” which is for accelerating calculation. Additionally, a special algorithm is developed for generating simulated images. The proposed method is validated by comparing measured displacements with those of simulated images, and proved to be much faster than the existing ones if the same accuracy is assumed. As the rigid displacements and the elastic displacements induced by deformation can be directly calculated by the affine transform, the proposed method can efficiently measure the full-field displacements of large displacement or finite deformation.

© 2006 Elsevier Ltd. All rights reserved.

Keywords: Affine transform; Coarse-fine search method; Nested fine search method; Digital image correlation method (DICM); Displacement measurement

1. Introduction

Measurement of surface deformation is very important in experimental mechanics and many optical methods related to the measurement, such as Moiré interferometry [1–4], shearography [5–7], electronic speckle pattern interferometry (ESPI) [8,9] and DICM [10–19,22–31], have been developed over the past years. As a powerful technique in photomechanics, DICM has attracted more and more attentions from many researchers in recent years. Perhaps this is due to its advantages including non-contact, real time, and full-field measurement, avoidance of tedious phase information, and interferometric fringe treatment.

The DICM was originally developed in early 1980s for experimental stress analysis [10,11]. During the following two decades, the potential of DICM for analyzing stress and deformation
behave was recognized. Over the past 20 years, the DICM has been improved significantly for reducing computation complexity and achieving high accuracy with much reduced computation cost. For example, a new implementation was proposed in [16,17] for increasing output accuracy. The implementation was made by incorporating a second order approximation of the deformation mapping. Other implementations such as Double-parameters method [18], Newton–Raphson iterative method [19–21], quasi-Newton method [22,23], self-adaptive search method [24–26] and gradient-based algorithm [27] have also been proposed in the two decades to speed up optimization looping and improve convergent performance. It should be noted that, when using the approaches mentioned above for deformation analysis, an initial value of displacement is, in general, required as initial conditions in the correlation procedure and the estimated displacement is then obtained by one of the following approaches: (1) search by a ‘coarse’ or ‘coarse-fine’ search correlation approach [11]; (2) obtain a sample point at which the displacement is smaller than the initial guess, or search for it using the ‘coarse-fine’ search correlation approach, after which the results from previous correlations are taken as initial guesses for neighboring subsets. Again, this might not ensure convergence during optimization if two neighboring subsets are too faraway and the deformation is relatively large, with the result that the displacement difference between the two subsets would be greater than ±3 pixels. The third approach has the advantage of accelerating the coarse-searching speed, especially in the case of large rigid-body displacement. However, it is hampered by lack of automatization and intelligence. Consequently, the search schemes described above are either inefficient or weak in automatization and intelligentized operation, there is a need to develop a faster, more automatic and more intelligentized coarse-fine search method. This is the motivation of this work.

In this paper, a novel coarse-fine search scheme for DICM is developed. This scheme is based on an affine transform, correlation analysis and a new technique of fine searching. Here the affine transform is employed to determine the approximate position of each pixel in the undeformed image, in contrast to previous studies in which the affine transform was used as a shape function [28]. The procedure of this search scheme is as follows: firstly, the affine parameter is calculated through the identification of several points \((x_i, y_i)\) \((i = 1, 2, \ldots, n, and n \geq 3)\) with distinct features before and after deformation; secondly, the approximate position of each point in the region of interest is then evaluated by the affine parameter; finally, a coarse-fine search is conducted around the approximate position to obtain the exact displacement. In addition, an algorithm for generating simulated images is presented. The algorithm has been validated by comparing the measured displacements with the actual displacements of simulated image, and is shown to be much faster than the existing coarse-fine search scheme if the same accuracy tolerance is used.

2. Principles

2.1. Affine transform

Affine transform is a mathematical technique commonly used in image processing for obtaining
The counterpart of one image to others. The affine transform can map lines or curves from one coordinate system to another and establish relationship of lines between systems. It is noted that, in previous studies, the affine transform represented by first-order shape functions was used to allow a combination of translation, rotation, shear and normal strains. The affine transform can also, however, be used in the DICM to determine the approximate strains. The affine transform can also, however, be used in the DICM to determine the approximate strains. The affine transform can also, however, be used in the DICM to determine the approximate strains.

The affine transform is composed of a linear transform and a translation transform. In 2D space the transform can be expressed as

\[
\begin{bmatrix}
x' \\
y'
\end{bmatrix} = \begin{bmatrix}
a_{11} & a_{12} \\
a_{21} & a_{22}
\end{bmatrix} \begin{bmatrix}
x \\
y
\end{bmatrix} + \begin{bmatrix}
\tau_x \\
\tau_y
\end{bmatrix}
\]  

(1)

where \( \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \) is a real matrix, \( \begin{bmatrix} \tau_x \\ \tau_y \end{bmatrix} \) stands for the translation vector. Eq. (1) is known as the so-called six-parameter affine transform, and can be further written as

\[
\begin{bmatrix}
x' \\
y'
\end{bmatrix} = \begin{bmatrix}
a_{11} & a_{12} & \tau_x \\
a_{21} & a_{22} & \tau_y
\end{bmatrix} \begin{bmatrix}
x \\
y
\end{bmatrix} + \begin{bmatrix}
1 \\
1
\end{bmatrix}
\]  

(2)

where \( A = \begin{bmatrix} a_{11} & a_{12} & \tau_x \\ a_{21} & a_{22} & \tau_y \end{bmatrix} \) represents “affine transform matrix”.

The inverse of Eq. (2) yields

\[
\begin{bmatrix}
x \\
y
\end{bmatrix} = \frac{1}{a_{11}a_{22} - a_{12}a_{21}} \begin{bmatrix}
a_{22} & -a_{12} & a_{12}\tau_y - a_{22}\tau_x \\
-a_{21} & a_{11} & a_{21}\tau_x - a_{11}\tau_y
\end{bmatrix} \begin{bmatrix}
x' \\
y'
\end{bmatrix}
\]  

(3)

where \( B = \frac{1}{a_{11}a_{22} - a_{12}a_{21}} \begin{bmatrix}
a_{22} & -a_{12} & a_{12}\tau_y - a_{22}\tau_x \\
-a_{21} & a_{11} & a_{21}\tau_x - a_{11}\tau_y
\end{bmatrix} \) is known as “inverse affine transform matrix”.

Furthermore, supposing \((x'_i, y'_i)\) to be the matching point of \((x_i, y_i)\) \(i = 1, 2, \ldots, n\), the affine transform model can be expressed in the general form as

\[
X' = AW
\]  

(4)

where

\[
X' = \begin{bmatrix}
x'_1 & x'_2 & \cdots & x'_n \\
y'_1 & y'_2 & \cdots & y'_n
\end{bmatrix}, \quad W = \begin{bmatrix}
x_1 & x_2 & \cdots & x_n \\
y_1 & y_2 & \cdots & y_n
\end{bmatrix}, \quad A = \begin{bmatrix}
a_{11} & a_{12} & \tau_x \\
a_{21} & a_{22} & \tau_y
\end{bmatrix}
\]

The optimized transform matrix \( A \) can be obtained by way of least-square method. The least-square solution to \( A \) exists if and only if \( WW^T \) is a full-rank matrix, and can be written as

\[
A = XW^T(WW^T)^{-1}
\]  

(5)

The least-square solution to the inverse affine transform matrix \( B \) can be obtained similarly.

2.2. Interpolation

It is well known that the digital speckle pattern obtained by a digitizer (also called ‘image board’ or ‘image grabber’ in some publications) is in discrete form. Intensity values are recorded as ‘pixels’. Due to the discrete nature of digital images, a pixel in an undeformed image may be mapped to a position between pixels in the deformed image. To locate accurately the position of subpixel, the cross-correlation (or other matching criteria, e.g. the least-squares correlation coefficient) must be evaluated at noninteger locations. On the other hand, a pixel in the origin image may also map to a subpixel location after the transform when generating a simulated image. Therefore, an interpolation method is required to reconstruct the gray-value at the subpixel location. Several kinds of interpolation formula are used in the literature, such as the bilinear interpolation method, the polynomial interpolation method and the bicubic spline interpolation method. In the present work, the bicubic polynomial interpolation method is used and can be expressed as a convolution operator with the kernel function

\[
\text{Cubic}(d) = \begin{cases} 
(d - 2)^2 + 1 & (d \leq 1) \\
[(5 - d)d - 8.0]d + 4.0 & (1 < d \leq 2)
\end{cases}
\]  

(6)

where \( d \) is the \( x \) or \( y \) component of the distance between sampled pixels and subpixel location. With this method, the gray-value of an arbitrary location \((x, y)\) is calculated by the formula

\[
g(x, y) = \sum_{i=1}^{4} \sum_{j=1}^{4} g(x_i, y_j) \text{Cubic}(dx_i)\text{Cubic}(dy_j)
\]  

(7)

where \( g(x_i, y_j) \) is the gray-value of the nearest \(4 \times 4\) sampled pixels around the subpixel location, \( dx_i \) and \( dy_j \) are the \( x \) and \( y \) components of the distance between the sampled pixels \((x_i, y_j)\) and the subpixel location, respectively.
The bicubic interpolation method implements a third order polynomial that allows both gray-values and their derivatives (up to second order) to be continuous within each subdomain under consideration. The bicubic interpolation approach is a well-established mathematical method and can be found in the literature [31].

2.3. Proposed coarse-fine search method

2.3.1. Nested fine search method

After the coarse correlation search, a ‘fine’ search carried out at subpixel level is employed when necessary. Here each of the approximate values of displacement is equal to adding a decimal value to the approximate value obtained in the coarse correlation search. This process is not suitable for integral pixel patterns. In this case an interpolation formula is necessary, as described in the previous section.

The existing fine search methods usually search for the best matching points by a given step in the x and y directions, which are denoted by x_step and y_step, respectively, in all the possible subsets of the deformed intensity pattern in a square subset of 1 × 1 pixels centered at the location determined by coarse search. The computation complexity of these fine search methods is proportional to (x_step⁻¹ + 1) × (y_step⁻¹ + 1). It can be easily seen that this scheme is time-consuming and expensive, as a great number of correlation calculations must be executed. For example, if a precision of 0.01 pixels in both directions is expected, it is necessary to execute the calculation 101 × 101 (=10,201) times for each sample pixel. Therefore, a new fine search method called “nested fine search method” is here developed to reduce computation complexity. This new search strategy is used throughout this work, and its scheme is described below.

For simplicity, suppose \( x_{\text{step}} = y_{\text{step}} = 10^{-n} \) pixel \((n = 1, 2, \ldots)\), then the new search scheme is as follows: firstly, search to the precision of 0.1 pixel for both directions in a square subset of 1 × 1 pixels centered at the location given by coarse search, and record the matching point as \((x'_0, y'_0)\); secondly, search to the precision of 0.01 pixel for both directions in a square subset of 0.1 × 0.1 pixels centered at \((x'_1, y'_1)\), and record the matching point as \((x'_2, y'_2), \ldots\). Finally, search to the precision of \(10^{-n}\) pixel for both directions in a square subset of \(10^{-n+1} \times 10^{-n+1}\) pixels centered at \((x'_{n-1}, y'_{n-1})\), and record the matching point as \((x'_n, y'_n)\). Then \((x'_n, y'_n)\) is the best matching location that is expected. The procedure of this nested fine method is shown schematically in Fig. 1. Using this scheme, \(n \times 11 \times 11 \) (=121n) calculations are required for a precision of \(10^{-n}\) pixel in both directions. It is obvious that computation complexity is considerably reduced except for the case when \(10^{-n}\) is equal to 0.1, where the number of calculation is the same for the two schemes. For instance, for the precision of 0.01 pixels in both directions, only 242 calculations are required for the new search scheme, which is 3% less than that of the previous scheme.

2.3.2. Coarse-fine search method based on affine transform

The proposed coarse-fine search method based on the affine transform and nested fine search scheme is described as follows: Firstly, select \(n\) sample points \((x_i, y_i)\) \((i = 1, 2, \ldots, n; n = 3\) in our analysis) with distinct features within the region of interest in the undeformed image. Then search for the corresponding matching point \((x'_i, y'_i)\) in the whole field or in the given regions of the deformed image. The given regions are rectangles visually dragged out in the client window of the self-developed correlated software around the approximate locations of the distinct feature points within the deformed image. Secondly, calculate the affine matrix \(A\) through the identification of these points by Eq. (5). The location \((x'_p, y'_p)\) in the deformed image corresponding to the points \((x_p, y_p)\) can then be evaluated through Eq. (2). Then the nearest location to \((x'_p, y'_p)\) at the integer-pixel level is approximately taken as \((x''_p, y''_p)\). Subsequently, conduct a coarse search in a rectangle subset of Width × Height pixels centered at \((x''_p, y''_p)\) to obtain the most matching location \((x'''_p, y'''_p)\). Finally, the nested fine
search is carried out in the square subset of $1 \times 1$ pixels centered at $(x_p', y_p')$ for the exact matching location $(x_m', y_m')$, and the displacement of each sample point can thus be obtained. The procedure described above is schematically shown in Fig. 2.

3. Methodology verification using simulated images

3.1. Generation of simulated images

Computer-simulated images are used for verification of the proposed coarse-fine search method due to the ease of controlling image features and the predicted information of deformation. However, since commonly used software cannot translate an image by subpixel displacement, it is necessary to develop a technique for generating simulated images at the subpixel level. A typical simulation method for such purpose was proposed in [28] based on the discrete Fourier transform. Its process is too complex to be used in practical analysis, although the method can avoid the introduction of unwanted phase errors associated with polynomial or B-spline interpolators. For efficiently generating simulated images, a simulated technique based on affine transform is developed and described below in detail. The technique is suitable and sufficiently accurate in comparison with the previous methods [28].

In the proposed method, we suppose the grayscale level of an arbitrary location $(x, y)$ is $g(x, y)$ for a given original image, then using Eq. (2), the grayscale level $g(x', y')$ of an arbitrary transformed location $(x', y')$ can be calculated by

$$
g(x', y') = \begin{cases} g(x, y), & x \geq 0 \text{ and } y \geq 0 \\ 0 \text{ or } 255, & x < 0 \text{ or } y < 0 \end{cases}$$

subject to

$$\begin{align*}
0 \leq x' &\leq \text{ceil}(a_{11}(n\text{Width} - 1) + \tau_x, a_{11}(n\text{Width} - 1) + a_{22}(n\text{Height} - 1) + \tau_y)) \\
0 \leq y' &\leq \text{ceil}(a_{22}(n\text{Height} - 1) + \tau_y, a_{21}(n\text{Width} - 1) + a_{22}(n\text{Height} - 1) + \tau_y)) \\
& a_{11} > 0, \quad a_{22} > 0
\end{align*}$$

where $n\text{Width}$ and $n\text{Height}$ are the width and height of the original image, respectively. The function $\text{ceil}(x)$ represents the smallest integer that is greater than or equal to $x$. The coordinates $(x, y)$ corresponding to $(x', y')$, can be determined from Eq. (3). When $(x, y)$ is not at the integer-pixel level, the interpolation method, Eq. (7), is used to reconstruct the gray-value at the subpixel level.

The algorithm above is suitable for generating the corresponding simulated image of the original image with given rigid-body translation or with preassigned deformation, such as rigid-body translation with $[u, v]^T = [\tau_x, \tau_y]^T$ or isotropy homogeneous distortion with displacement $[u, v]^T = [2x_1x + \tau_x, x_2y + \tau_y]^T$ and displacement gradient $[u_x, v_y]^T = [a_{11} - 1, a_{22} - 1]^T$. These two typical deformation configurations are employed to verify the proposed coarse-fine method in the following section.

3.2. Verification using simulated images

A typical gray pattern is used as the original image with a chosen resolution of $128 \times 128$ and 8-bit per pixel. Simulated images of this original image are used to verify the accuracy and the computational efficiency of the present coarse-fine search method. Two typical deformation configura-
tions are used to generate image pairs before and after deformation: (1) rigid-body translation with displacement $[u, v]^T = [x, 0]^T$, and (2) uniaxial tensile with displacement $[u, v]^T = [z_{11} x, 0]^T$ and displacement gradient $[u_x, v_y]^T = [z_{11} - 1, 0]^T$. The present coarse-fine search method and the previous search scheme are employed to determine the integer-pixel and subpixel displacements. A comparison of these calculated values with the preassigned values reveals the accuracy of these schemes. Further, a comparison of calculating times between the present scheme and the previous one demonstrates the computational efficiency of the present scheme. For each image pair, a region of $102 \times 102$ pixels in the original image is investigated. There are 324 points of interest with fixed positions in this region. All computations are completed under the same conditions of PIV-2.4G mainframe with 256M memory, and the correlated subset size is $21 \times 21$ pixels.

3.2.1. Accuracy comparison with preassigned displacement

Accuracy of the proposed method is verified and compared with the existing methods in this section. The verification and comparison are made using the simulated undeformed images with preassigned rigid-body displacement $[u, v]^T = [0.001 - 1.0, 0]^T$ pixels and deformed images with both preassigned displacement $[u, v]^T = [1.05 x, 0]^T$ pixels and displacement gradient $[u_x, v_y]^T = [0.05, 0]^T$.

For the translated case, the measured mean displacements and standard deviations of the 324 points in the original image are calculated, and the result is listed in Table 1. It can be seen from Table 1 that the measured mean displacement is almost identical with the preassigned value, except when the preassigned displacement $u$ is equal to 0.25 and 0.75 pixels, where highest reconstruction error caused by intensity interpolation occurs [31]. Therefore, the measured mean values at these two positions are not taken into account when determining the linear fit of the mean values, and the result is shown in Fig. 3(a). Again, it can be seen from Fig. 3(a) that a good agreement is observed between the linear fit data and the preassigned displacements.

For the uniaxial deformed case, since the true displacement is constant along vertical lines in the image, the average is taken along the vertical direction, where 18 data points are available for each line, resulting in a total of 18 mean values calculated, as shown in Table 2. Although the differences between the mean values and the preassigned values in the deformed case are always relatively larger than those in the translated case, the mean values are thought to be sufficiently accurate enough to be the initial displacement values, accounting for shape changing in the deformed case and the maxi-

### Table 1

<table>
<thead>
<tr>
<th>Exact value</th>
<th>Mean value</th>
<th>Difference</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.001</td>
<td>0.00099</td>
<td>0.00001</td>
<td>0.00004</td>
</tr>
<tr>
<td>0.005</td>
<td>0.00494</td>
<td>0.00006</td>
<td>0.00023</td>
</tr>
<tr>
<td>0.01</td>
<td>0.00991</td>
<td>0.00009</td>
<td>0.00039</td>
</tr>
<tr>
<td>0.025</td>
<td>0.02472</td>
<td>0.00028</td>
<td>0.00090</td>
</tr>
<tr>
<td>0.05</td>
<td>0.04911</td>
<td>0.00089</td>
<td>0.00176</td>
</tr>
<tr>
<td>0.075</td>
<td>0.07288</td>
<td>0.00212</td>
<td>0.00238</td>
</tr>
<tr>
<td>0.1</td>
<td>0.09590</td>
<td>0.00410</td>
<td>0.00295</td>
</tr>
<tr>
<td>0.25</td>
<td>0.22311</td>
<td>0.02689</td>
<td>0.00477</td>
</tr>
<tr>
<td>0.5</td>
<td>0.50144</td>
<td>0.00144</td>
<td>0.00942</td>
</tr>
<tr>
<td>0.75</td>
<td>0.77673</td>
<td>0.02673</td>
<td>0.00485</td>
</tr>
<tr>
<td>1.0</td>
<td>1.00000</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Unit: pixel.

Fig. 3. Verification of the algorithm under (a) preassigned rigid-body translation and (b) homogeneous deformation.
The difference is less than 0.05 pixels. Moreover, as the true displacement is also proportional to the $x$ location, a linear fit of the mean values is conducted, as shown in Fig. 3(b). In a similar manner, a good agreement is observed between the linear fit data and the preassigned displacements.

### 3.2.2. Comparison of accuracy and computation time between coarse-fine search methods

Simulated images with rigid-body translation are generated to compare the accuracy and computation time of the presented search scheme with the previous method. Four typical cases each of which uses the same minimum step are considered. In each case, a region of $102 \times 102$ pixels with 324 points of interest is investigated, and the data obtained are listed in Table 3.

It can be seen from Table 3 that the accuracy of the present scheme is identical to that of the previous scheme when the same minimum step is used. The computation time for the presented search scheme, however, is much less than that of the previous scheme except for the case 3. In case 3, same computation time is required for the fine search of the two schemes when the same minimum step (0.1 pixels here) is used. While much more time is required for the present coarse search scheme as an additional calculation induced by the affine transform is required when the same coarse-searching region size is used. It should be mentioned that both two search processes include two major stages: fine search and coarse search. The time spent in each of these two stages may be different from one method to another. The computing time listed in Table 3 is the total time spent in the whole process. Therefore, it is recommended that position predetermination using affine transform should be avoided if the exact displacement is less than 3 pixels. In cases 1 and 2, although computation time for the coarse search is longer, the proposed coarse-fine search strategy is still much faster than the previous search method. The nested fine search scheme can considerably increase the calculating speed. In case 4, affine transform permits small coarse-searching region size, thus can significantly increase the calculating speed during the searching process. This is particularly true when the translation is very large even if same minimum step not less than 0.1 pixels of fine search is used, indicating that the computation time of the fine search for these two schemes is usually the same. As a result, the proposed coarse-fine search method has an advantage over

#### Table 2

<table>
<thead>
<tr>
<th>$x$</th>
<th>Exact value</th>
<th>Mean value</th>
<th>Difference</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>0.55</td>
<td>0.57756</td>
<td>0.02756</td>
<td>0.00877</td>
</tr>
<tr>
<td>17</td>
<td>0.85</td>
<td>0.88489</td>
<td>0.03489</td>
<td>0.00979</td>
</tr>
<tr>
<td>23</td>
<td>1.15</td>
<td>1.12400</td>
<td>0.02600</td>
<td>0.00745</td>
</tr>
<tr>
<td>29</td>
<td>1.45</td>
<td>1.45789</td>
<td>0.00789</td>
<td>0.01765</td>
</tr>
<tr>
<td>35</td>
<td>1.75</td>
<td>1.79028</td>
<td>0.04028</td>
<td>0.01501</td>
</tr>
<tr>
<td>41</td>
<td>2.05</td>
<td>2.02272</td>
<td>0.02728</td>
<td>0.02136</td>
</tr>
<tr>
<td>47</td>
<td>2.35</td>
<td>2.30361</td>
<td>0.04639</td>
<td>0.01319</td>
</tr>
<tr>
<td>53</td>
<td>2.65</td>
<td>2.66778</td>
<td>0.01778</td>
<td>0.01164</td>
</tr>
<tr>
<td>59</td>
<td>2.95</td>
<td>2.97494</td>
<td>0.02494</td>
<td>0.01244</td>
</tr>
<tr>
<td>65</td>
<td>3.25</td>
<td>3.21761</td>
<td>0.03239</td>
<td>0.00937</td>
</tr>
<tr>
<td>71</td>
<td>3.55</td>
<td>3.56983</td>
<td>0.01983</td>
<td>0.01148</td>
</tr>
<tr>
<td>77</td>
<td>3.85</td>
<td>3.88572</td>
<td>0.03572</td>
<td>0.01285</td>
</tr>
<tr>
<td>83</td>
<td>4.15</td>
<td>4.11744</td>
<td>0.03256</td>
<td>0.02101</td>
</tr>
<tr>
<td>89</td>
<td>4.45</td>
<td>4.42444</td>
<td>0.02556</td>
<td>0.00807</td>
</tr>
<tr>
<td>95</td>
<td>4.75</td>
<td>4.79250</td>
<td>0.04250</td>
<td>0.01221</td>
</tr>
<tr>
<td>101</td>
<td>5.05</td>
<td>5.04700</td>
<td>0.00300</td>
<td>0.01420</td>
</tr>
<tr>
<td>107</td>
<td>5.35</td>
<td>5.31011</td>
<td>0.03989</td>
<td>0.01180</td>
</tr>
<tr>
<td>113</td>
<td>5.65</td>
<td>5.66144</td>
<td>0.01144</td>
<td>0.01120</td>
</tr>
</tbody>
</table>

Unit: pixel.

#### Table 3

<table>
<thead>
<tr>
<th>Case</th>
<th>Search scheme</th>
<th>Exact value</th>
<th>Mean value</th>
<th>Difference</th>
<th>Standard deviation</th>
<th>Minimum step</th>
<th>Coarse-searching region size</th>
<th>Computation time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>New</td>
<td>0.005</td>
<td>0.00494</td>
<td>0.00006</td>
<td>0.00023</td>
<td>0.001</td>
<td>3 \times 3</td>
<td>158.735</td>
</tr>
<tr>
<td></td>
<td>Previous</td>
<td>0.005</td>
<td>0.00494</td>
<td>0.00006</td>
<td>0.00023</td>
<td>0.001</td>
<td>3 \times 3</td>
<td>121,668.967</td>
</tr>
<tr>
<td>2</td>
<td>New</td>
<td>0.01</td>
<td>0.01</td>
<td>0</td>
<td>0</td>
<td>0.01</td>
<td>3 \times 3</td>
<td>106.609</td>
</tr>
<tr>
<td></td>
<td>Previous</td>
<td>0.01</td>
<td>0.01</td>
<td>0</td>
<td>0</td>
<td>0.01</td>
<td>3 \times 3</td>
<td>1226.781</td>
</tr>
<tr>
<td>3</td>
<td>New</td>
<td>1.0</td>
<td>1.0</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
<td>3 \times 3</td>
<td>54.641</td>
</tr>
<tr>
<td></td>
<td>Previous</td>
<td>1.0</td>
<td>1.0</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
<td>3 \times 3</td>
<td>14.969</td>
</tr>
<tr>
<td>4</td>
<td>New</td>
<td>10.0</td>
<td>10.0</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
<td>3 \times 3</td>
<td>106.531</td>
</tr>
<tr>
<td></td>
<td>Previous</td>
<td>10.0</td>
<td>10.0</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
<td>21 \times 21</td>
<td>1269.032</td>
</tr>
</tbody>
</table>

Unit: pixel.
the previous search scheme in view of computing time. The proposed method can be used as a real-time processing tool with particular speed requirements for analyzing problems with larger displacements.

4. Conclusions

A novel coarse-fine search scheme for DICM is developed by introducing affine transform and a nested fine search strategy into the previous DICM. Numerical investigation shows that the proposed method has three features: (1) the nested fine search method can achieve higher processing speed than that from the previous coarse-fine search scheme if the same accuracy is assumed; (2) as the approximate rigid displacement and elastic displacements induced by deformation can be directly and automatically determined using affine transform, the approximate location of objective images corresponding to the rigid displacement and elastic displacements can thus be determined. As a result, the process for repeatedly searching for the rigid displacement of each sample pixel, which is necessary in the previous search methods, can be omitted. A relatively smaller searching region size and a smaller number of calculations for coarse search are then achieved, and the proposed scheme becomes more automated and more intelligent than previous schemes; (3) a relatively smaller coarse-searching region size can decrease the probability of misidentification and increase the quality of identification. As a result, good agreement is achieved between the measured data and simulation results. In conclusion, the approach is much faster than the existing ones if the same accuracy is assumed, thus has an advantage over the previous scheme in view of computing time. In particular, it is suitable to be used as a real-time processing tool with particular speed requirements. It concludes that a combination of DICM with iterative methods and affine transform can result in a more accurate, more automated and more intelligent measurement technique for measuring full-field displacements of large displacement and finite deformation.

Acknowledgements

This work was financially supported by the National Natural Science Foundation of China (Nos. 10232030 and 10472082) and the Australian Research Council (No. DP0209487). The supports are fully acknowledged.

References


