Tomasz MARCINIAK, Zbigniew LUTOWSKI, Sławomir BUJNOWSKI, Dariusz BOROŃSKI
University of Technology and Life Sciences, Bydgoszcz
Tomasz GIESKO
Institute for Sustainable Technologies – National Research Institute, Radom

FAST DISPLACEMENT ANALYSIS BY DIGITAL IMAGE CORRELATION (DIC) BASED ON MULTI-PROCESSOR GRAPHIC CARDS (GPU)

Key words

Digital image correlation (DIC), multi-processor graphic cards (GPU), whole-field displacement and strain analysis.

Summary

The paper presents a method of fast displacement and strain analysis based on digital image correlation (DIC) and multi-processor graphic cards (GPU). The basic assumption for the discussed displacement and strain measurement method under time variable loads was that high measurement sensitivity by simultaneously minimising measurement time consumption was possible. For this purpose special computing procedures based on multi-processor graphic cards (GPU) were developed that significantly reduced the total time of displacement and strain analysis.

Introduction

Difficulties in investigations of strains in technical objects are essential problem in experimental mechanics of solids. One of these is a strain measurement in fatigue cracking zones in structures subject to cyclically reversed loading. However, an important problem in these investigations is necessity to obtain measurements in the dynamic mode. It is particularly important in whole-field strain measurement techniques. In many cases these are measurement methods based on the optical techniques, in which optoelectronic systems are used. Therefore, it is important to reduce the analysis time of images captured by digital cameras and analyse them with the use of numerical procedures. In this paper, a method for fast, whole-field displacement and strain distribution analysis in on-line mode is presented in which digital image correlation procedures are used for displacement measurement.

Whole-field strain measurement with the use of digital image correlation is an exceptionally attractive choice in the mechanics of solids [1, 2]. However, its practical application, especially for an analysis of small strain values, is strongly restricted. It is mainly caused by technological limitations, especially for materials with high stiffness, for example, steel. Measurement of small displacements requires the use of very high-resolution digital cameras, even when sub-pixel interpolation methods are used. It involves the necessity of transmission and processing of huge amount of data in a very short time. Cyclical loads involve a significant increase in the amount of data to be processed with further reduction of the analysis time.

In the paper, example of displacement and strain analysis in the fatigue-cracking zone in specimens taken from aeroplane structures is presented.

2. Theoretical background

The principle of operation of image correlation methods is comparison of specimen surface images before and after exposing the object to load. For object lighting white light or laser light (spot methods) may be used [3, 4, 5, 6]. Displacements of surface characteristic points allow the determination of the displacement values within the analysed area. The sensitivity of this method depends on the parameters obtained using image observation methods, such as dimensions of the observation field or the image geometric resolution.

In the case of many objects, measurements can be realised with the use of their natural roughness; therefore, it is not necessary to specially prepare the surface to be analysed. Another method to obtain random points on the sample surface is marking them by means of special surface preparing techniques, including painting. Image surface points are recorded before and after exposing the object to strains. This enables the correlation of these images based on the intensity of each pixel recorded by a CCD matrix or a different type of image sensor. The usage of interpolation functions may increase accuracy as compared to a strictly digital analysis carried out for a pixel grid. Moreover, methods of artificial intelligence, including genetic algorithms, may be used for the analysis of the image spot.

Analysis of displacements based on the assessment of the cross correlation coefficient *C* is one of the basic solutions used in the image cross correlation methods. The cross correlation coefficient is determined for a chosen point of the environment, *P*[*x*, *y*], as follows:

$$C(u,v)=\sum_{i=a}^{b}\sum_{j=c}^{d}\left[I_n(x_i,y_j)I_m(x_i,y_j)\right] \tag{1}$$

for a standardised version

$$C(u,v)=\sum_{i=a}^{b}\sum_{j=c}^{d}\left[\frac{I_n(x_i,y_j)I_m(x_i,y_j)}{\bar{f}\cdot\bar{g}}\right] \tag{2}$$

for a standardised version, where $\bar{f}$ and $\bar{g}$ are, respectively:

$$\bar{f}=\sqrt{\sum_{i=a}^{b}\sum_{j=c}^{d}\left[I_n(x_i,y_j)\right]^2} \tag{3}$$

$$\bar{g}=\sqrt{\sum_{i=a}^{b}\sum_{j=c}^{d}\left[I_m(x_i,y_j)\right]^2} \tag{4}$$

where:

a, *b* – initial and final values of the correlated image coordinate *x* index,

c, *d* – initial and final values of the correlated image coordinate *y* index,

$I_n(x_i, y_j)$ – the first image intensity in the point with coordinates *x y*, recorded in the phase of loading,

$I_m(x_i, y_j)$ – the second image intensity in the point with coordinates *x y*, recorded in the phase of loading.

The value of the coefficient *C* can change in the range from 1 to 0. When *C* = 1, the images of the environment of the point *P*, recorded in phases *n* and *m*, are entirely consistent and when *C* = 0 they are entirely different.

In order to define a displacement of a chosen point *P* based on images recorded in the two phases of loading *n* and *m*, sub-area O covering point *P* is separated from image *n* and *m* is displaced in relation to the analogue area in image *n*. The value of cross correlation coefficient *C* is calculated for each location of sub-area On.

2. Fast Digital Image Correlation Method Based on GPU

Due to low efficiency of standard functions to determine the coefficient *C*, e.g. those available in OpenCV library, an original algorithm of correlative search of the image similar areas has been developed. The implementation was performed by means of CUDA library in version 3.2, facilitated by NVidia Company. Since there are two known ways of obtaining a resultant matrix of the pattern adjustment to the searched image (correlation matrix), it is necessary to make a choice in the initial phase, between the method based on Fourier transforms used by Open CV library and the method to directly determine the values of the correlation function. It has been assumed that the pattern area dimensions are: from 20×20 to 40×40 points. For such dimensions the approach using transforms loses is efficient because of a higher number of successive stages of computing. Therefore, it was decided to carry out our own implementation of the algorithm determining the correlation function according to the version directly defined by equation 2.

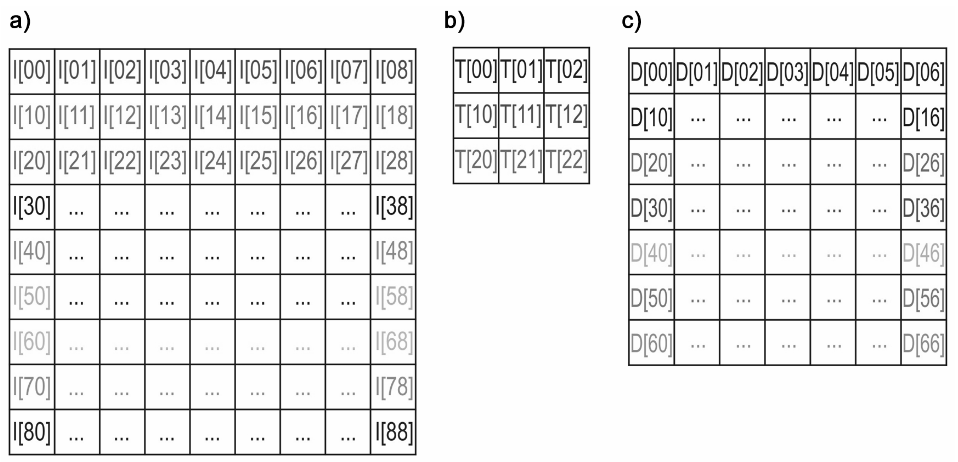
a)
I[00] I[01] I[02] I[03] I[04] I[05] I[06] I[07] I[08]
I[10] I[11] I[12] I[13] I[14] I[15] I[16] I[17] I[18]
I[20] I[21] I[22] I[23] I[24] I[25] I[26] I[27] I[28]
I[30] ... ... I[38]
I[40] ... ... I[48]
I[50] ... ... I[58]
I[60] ... ... I[68]
I[70] ... ... I[78]
I[80] ... ... I[88]
b)
T[00] T[01] T[02]
T[10] T[11] T[12]
T[20] T[21] T[22]
c)
D[00] D[01] D[02] D[03] D[04] D[05] D[06]
D[10] ... ... D[16]
D[20] ... ... D[26]
D[30] ... ... D[36]
D[40] ... ... D[46]
D[50] ... ... D[56]
D[60] ... ... D[66]

Fig. 1. Matrixes of searched image I (a), looked for pattern T (b), resultant D (c) (source: authors)

Due to the fact that loading data into the global graphic card memory is one of the most time consuming operations affecting the general efficiency of the implemented algorithm, it was decided to carry out calculation threads connected with one line of searched image **I** in a single block. For this line, we have performed all combinations connected with it of multiplications by the image of pattern **T**. Therefore it is possible to avoid loading the same data for image **I** at the cost of reloading the pattern data (the pattern area is assumed to be smaller than the searched one). Since the data concerning points of the one image line represents successive memory cells, physically adjacent to each other, certain number *n* of the data reading operations (typically including 128 bytes) can be combined into a single operation. Examples of matrixes corresponding to the searched fragment of image **I** of size 9 and pattern **T** of size 3 are shown in Figure1.

Thus, operations of partial sums of products belonging to different displacements of the pattern image are carried out by a single block of threads, which is shown in Figure 2.

For instance, in Figure 3, the calculations performed by the third block of threads (connected with the 3rd line of matrix **I**) have been demonstrated in the successive phases of the algorithm. In the first step, thread 1 calculates the sums of products of the first pattern line by corresponding to it image points, for displacement D(2,0), simultaneously, these sums are calculated by thread 2, for displacement 2D(2,1), and sums for maximal, possible for X axis, displacement D(2,6), are calculated for exemplary parameters by the last thread. The following step of the algorithm involves loading the next line of pattern T (points T(1,0)..91,2)) and multiplying it by the same line of the image data, thus obtaining a part of the products sum necessary for the determination of displacement (1,0) for thread 1, displacement(1,1) for thread 2 etc.

```
a)                    b)                    c)
D[11]=I[11]*T[00]     D[11]=D[11]+          D[11]=D[11]+
           +                I[21]*T[10]           I[31]*T[20]
      I[12]*T[01]                 +                     +
           +                I[22]*T[11]           I[32]*T[21]
      I[13]*T[02]                 +                     +
                            I[23]*T[12]           I[33]*T[22]
```

Fig. 2. Example obtaining the correlation function values between image **I** and pattern **T** for point (1,1) of a resultant matrix (source: authors)

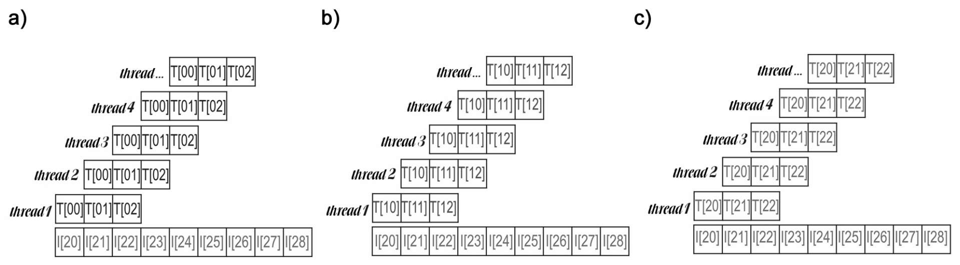

Fig. 3. Calculations performed by the block of threads connected with the third line of image **I** (source: authors)

The first and last z of image **I**, where z means the size of matrix **T**, are special cases. For instance, Figure 4 shows how the block of threads works for second line of image **I**. In this case, multiplication of the third line of pattern **T** is not performed, because such a displacement of the pattern would mean crossing the range of searched image **I**.

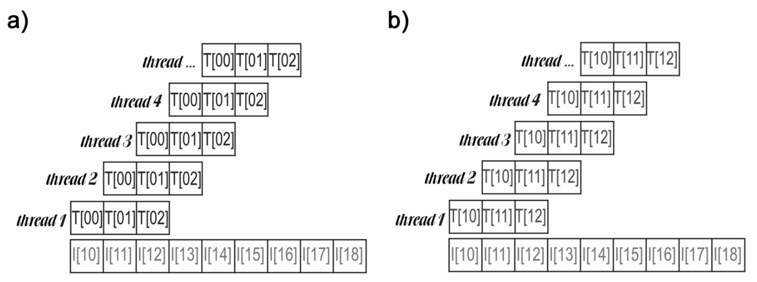

Fig. 4. Calculations performed by a sample block of threads – the image initial line (source: authors)

Since there is no possibility of synchronising particular blocks of threads with each other, it is necessary to use the operation of atomic (indivisible) addition in order to avoid potential errors of indirect recording of results. Such functionality is available for floating point data type, merely in the new CUDA (version 2.0-platform FERMI –atomicAdd). Tests were carried out comparing operational speed of algorithms using atomicAdd with a normal addition operation (this version returned results burdened with resulting from the collision of reading operations with the recording ones). These tests revealed a decrease in the operation speed by less than 2%. In order to ensure maximum grouping coefficient of the reading and recording operations, the allocation of global memory blocks GPU was performed by means of function cuMemAllocPitch providing a proper arrangement (adjustment) of the allocated memory blocks within the GPU address space.

For this version of the algorithm tests were carried out comparing the performance with its possible maximum optimised equivalent, taken from OpenCV library. Thus, the developed implementation, computing the functions of correlation without normalisation on GPU was compared with a multi-thread OpenCV implementation on CPU and GPU. Tests were carried out for a constant ratio of pattern area T in relation to searched image I, being 1:3 when both areas were square. The results of comparison tests are shown in Figure 5 where “CPU” is the algorithm with simplest implementation performed on CPU (one thread), “GPU” is the tested implementation of algorithm performed on GPU, “CPU-CV” represents an algorithm using OpenCV library in a multithread manager, “GPU-CV” is utilisation of Open CV library in GPU version.

The authors’ implementation of GPU is more efficient than the other solutions for T pattern dimensions not exceeding 65×65 points. For T pattern dimensions 30×30 points the computation speed achieved was ten times faster. Further optimisation of the algorithm’s speed was accomplished based on analysing the theoretical load of GPU (Tab. 1) The usage of GPU resources using the algorithm for the pattern dimension 15×15 points, reached 18%. The small number of threads triggered within a single block was the cause of limited usage of GPU. The documentation available from NVIDIA shows that it is necessary to aim at approximately 60% of GPU occupancy which, in case of the developed procedures, meant the pattern dimensions is supposed to be 60×60 points.

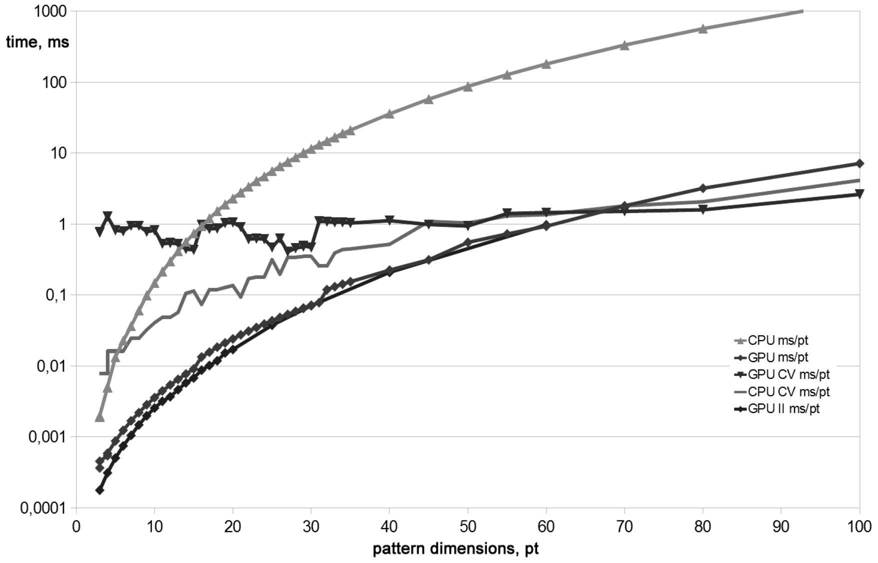

Fig. 5. Results of the correlation algorithm efficiency tests on GPU: "CPU" – algorithm with simplest implementation performed on CPU (one thread), "GPU" – the tested implementation of algorithm performed on GPU, "GPUII" – the tested implementation of algorithm performed on GPU, "CPU-CV" – algorithm using OpenCV library in a multithread manager, "GPU-CV" – utilisation of Open CV library in GPU version (source: authors)

Thus, an increase in GPU resources occupancy was obtained through a simultaneous computing of product sums for a few lines in a single block of threads. An optimal number of simultaneously computed lines ranges from 7 for T size = 14x14 points to 3 for T = 60. For bigger sizes of T, increasing the number of threads per block (with increased complexity of the algorithm used) did not improve the results. The effects of GPU resources occupancy optimization are demonstrated in Figure 6. Additionally, increasing the number of registers used by the block (with increased complexity of the algorithm) already caused a drop in GPU resources occupancy.

The above operations caused an increase of the computing speed for T sizes of 20 points by up to 100%. The effects of how this version of the algorithm works is presented in Figure 5 as GPU II.

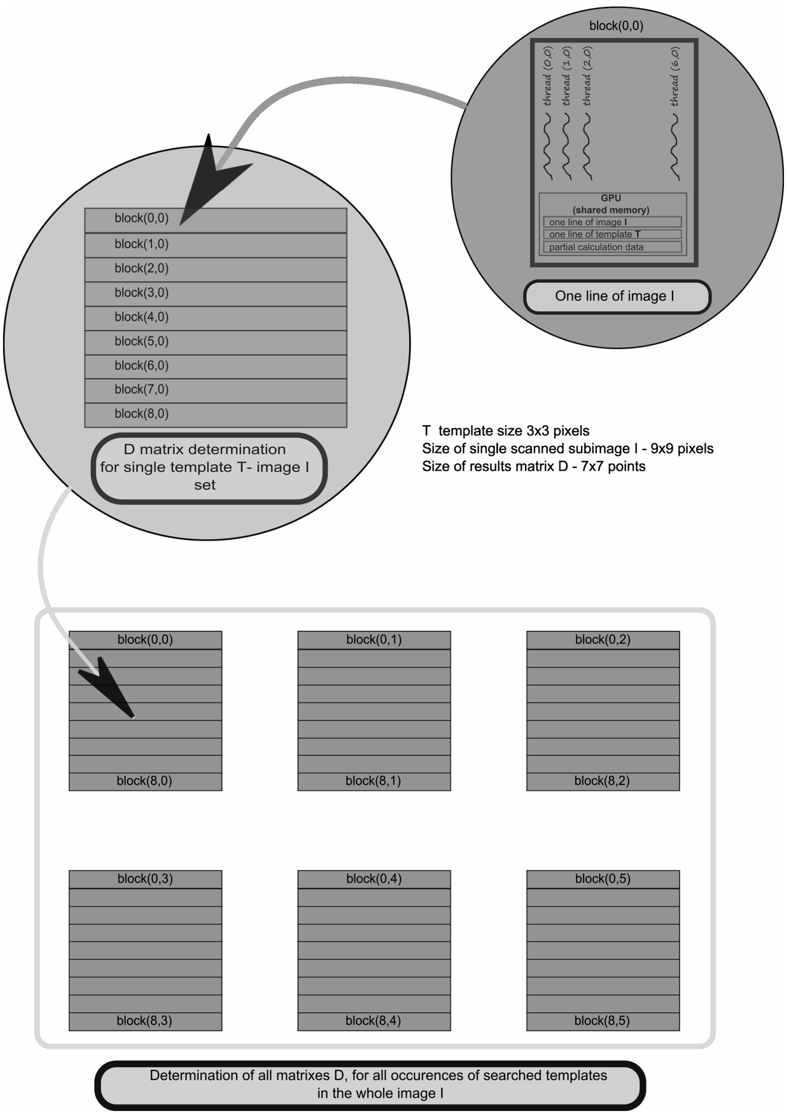

Fig. 6. Organisation of the algorithm to determine the correlation function, searching for the displacements of the pattern matrixes with dimensions equal to 3 columns per 2 lines (source: authors)

By copying a group of blocks of threads which computes the best match for a single area in the searched image, a simultaneous calculation of these matches became possible for the whole matrix of the searched areas (earlier defined by the user). The possibility of defining both blocks and threads performing the assigned program (kernel) in the form of a multi-dimension grid appears to be very helpful. As in Figure 6, inside blocks, threads are organised bi-dimensional, i.e. the first dimension corresponds to columns of determined matrix D, the second dimension corresponds to the analysed line of image I (optimisation increasing the number of threads per block). The best method to organise a block would be a three-dimensional organisation. Unfortunately, for the available version of CUDA library, an application of only two dimensions was possible, where the first dimension is the line of analysed image I, and the second dimension is represented by successive processed areas of the image with the x and y coordinates of a given area appropriately coded.

The last phase of the algorithm optimisation involved the use of texture memory for storing the pattern and searched images data. The main advantage of the textures memory is the so-called 2D cache – cache memory taking into consideration the image mutual adjacency of both horizontal and vertical axes points. Other advantages include hardware conversion between the integer and floating-point types and a possibility of hardware, bilinear interpolation of the values between the image points. By using the memory of textures, the algorithm's efficiency was improved by approximately 25%, regardless of pattern area T.

3. Example results of strain analysis in fatigue crack zone

The developed digital procedure for correlation of images has been used for example for displacement analysis in the method of fatigue crack propagation testing in aeroplane riveted joints.

In Figure 7, a specimen made of aluminium alloy 2024-T3 with a propagated crack, which was exposed to constant amplitude of nominal stress and the stress ration R = 0, is presented.

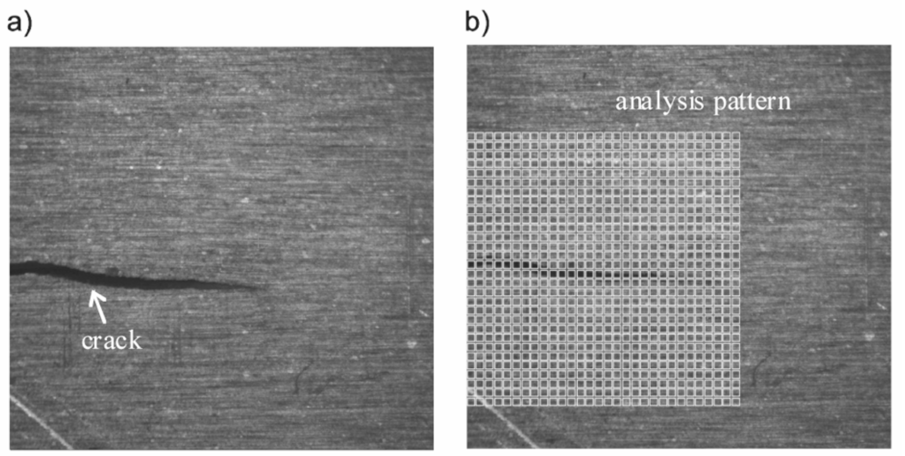

Fig. 7. Specimen with a crack (a) and crack image with analysis pattern (b) (source: authors)

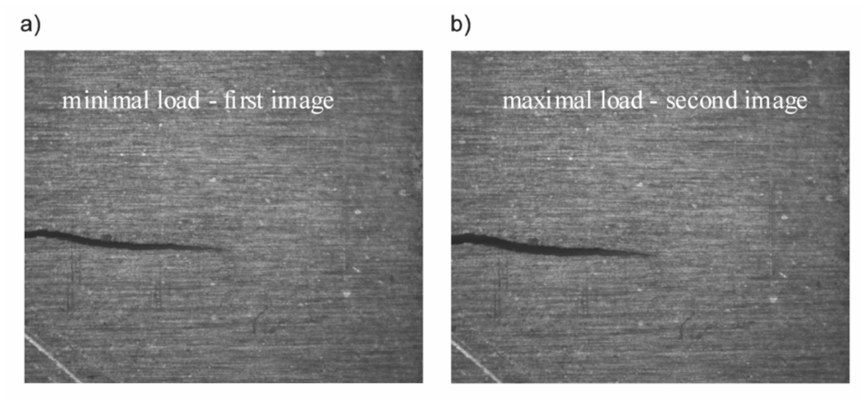

Fig. 8. Images of the specimen surface in chosen phases of the load cycle: a) minimal value of loading cycle, b) maximal value of loading cycle (source: authors)

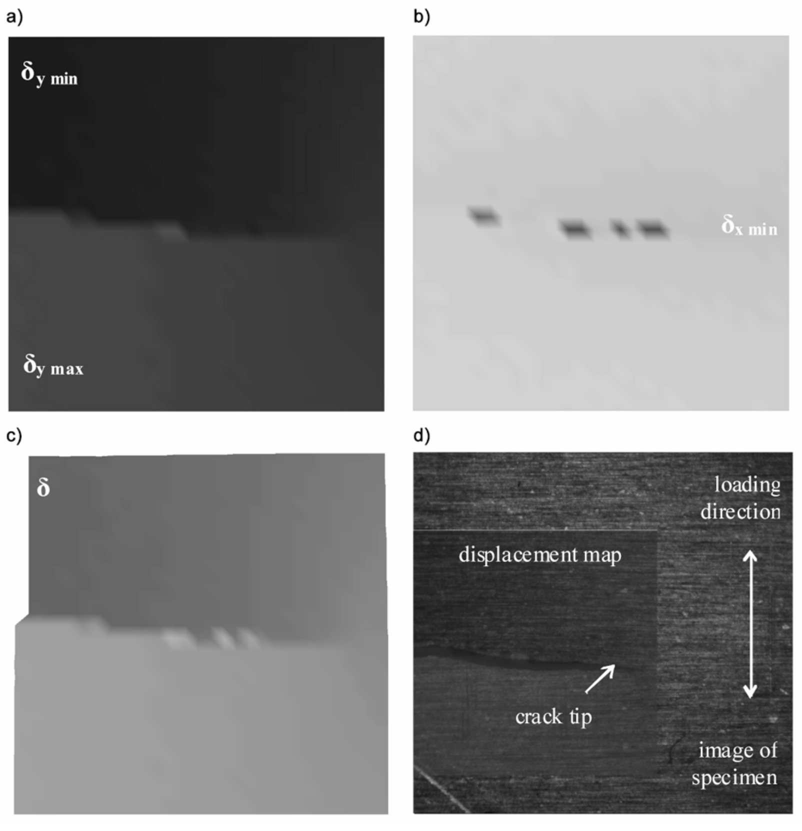

Fig. 9. Distributions of displacements within the crack zone: a) in loading direction δ_v, b) perpendicular to loading direction δ_u, c) resultant $\delta = \sqrt{\delta_v^2 + \delta_u^2}$, d) displacement map in loading direction against the background of specimen image (source: authors)

A digital video camera with an image resolution of 2448×2050 pixels and a set of lens with 50mm area of view was used for the sample image recording. The sample surface image was recorded on a PC equipped with GPU card NVIDIA GTX480 by means of a Giga Ethernet card with a maximum frequency of 17Hz. Measurement sensitivity s = 0.0025 mm was obtained for the applied resolution of the video camera, the matrix size, and lens zoom. In Figure 8, the sample images recorded with the use of dual-camera vision system for fatigue monitoring [7] is shown in chosen phases of a time variable load cycle. On this basis, distributions of displacements within the crack environment are demonstrated in Figure 9.

Conclusions

The presented method enables fast displacement and strain analysis in objects exposed to time variable loads. The results of measurement of displacements and deformations can be used in carrying out studies on fatigue and fracture mechanics of materials and structures, in cases in which information about the current state of displacement is used to control the test process. Moreover, in the case of displacement analysis in decreased areas of interest, reducing the time of the analysis offers the possibility of applying the developed method in fatigue life testing with controlled strain value, including local strain values.

By applying the digital image correlation technique supported by GPU technology and the use of modern optoelectronic and IT solutions, the image analysis algorithms developed for the equipment configuration used in this work was able to determine the strain distribution in an 2448×2050 pixels image, with the size of a searched patterns matrix of 30×30 elements and the pattern size of 20×20 pixels, in less than 0.2sec. Further reduction of the analysis time is possible for GPU cards with a higher number of cores.

Scientific work executed within the Strategic Programme "Innovative Systems of Technical Support for Sustainable Development of Economy" within Innovative Economy Operational Programme.

References

1. Pilch A., Mahajan A. and Chu T.: (2004) Measurement of whole-field surface displacements and strain using a genetic algorithm based intelligent image correlation method, Journal of Dynamic Systems, Measurement, and Control, v. 126, 3, 479–488.

2. Lagattu F., Brillaud J. and Lafarie-Frenot M.-C.: (2004) High strain gradient measurements by using digital image correlation technique, Materials Characterization 53, 17–28.
3. Schmidt T. and Tyson J.: (2003) Dynamic Strain Measurement Using Advanced 3D Photogrammetry, Proceedings of IMAC XXI, Kissimmee.
4. www.correlatedsolutions.com.
5. www.gom.com.
6. Boroński D.: (2007) Strain and stress analysis methods in fatigue of materials and structures, Wydawnictwo Naukowe Instytutu Technologii Eksploatacji – PIB, Bydgoszcz–Radom (in polish).
7. Giesko T.: (2012) Dual-camera vision system for fatigue monitoring, Materials Science Forum, vol.726, 226–232.
8. Marciniak T., Lutowski Z., Bujnowski S., Boronski D., Czajka P.: (2012) Dual-Band Experimental System For Subsurface Cracs Testing, Materials Science Forum, vol. 726, 222–225.

Recenzent:
Paweł PYRZANOWSKI

Szybka analiza przemieszczeń z wykorzystaniem cyfrowej korelacji obrazu (DIC) oraz wieloprocesorowych kart graficznych (GPU)

Słowa kluczowe

Cyfrowa korelacja obrazu (DIC), wieloprocesorowa karta graficzna (GPU), analiza przemieszczeń i naprężeń w pełnym polu obserwacji.

Streszczenie

W artykule zaprezentowano metodę szybkiej analizy przemieszczeń i naprężeń z wykorzystaniem cyfrowej korelacji obrazu (DIC) oraz wieloprocesorowych kart graficznych (GPU). Podstawowym założeniem omawianej metody pomiarów przemieszczeń i naprężeń w warunkach zmiennych obciążeń było osiągnięcie wysokiej czułości pomiarowej przez minimalizację czasu pomiaru. W tym celu opracowano specjalne procedury przetwarzania wykorzystujące wieloprocesorową kartę graficzną (GPU), w znaczącym stopniu redukujące całkowity czas procesu analizy przemieszczeń i naprężeń.