Paper 57

©Civil-Comp Press, 2012
Proceedings of the Eighth International Conference
on Engineering Computational Technology,
B.H.V. Topping, (Editor),
Civil-Comp Press, Stirlingshire, Scotland

Block Matching Algorithms for Load Test Evaluation

G. Almeida[1,2], F. Melicio and J. Fonseca[1]
[1]Faculty of Science and Technology, Monte da Caparica, Portugal
[2]Lisbon Superior Engineering Institute, Portugal

Abstract

The traditional methodology used for displacement measurements in civil engineering load tests requires a considerable amount of equipment and a very complex installation procedure. In this paper, the authors used real images taken during a load test in order to measure by image processing techniques the displacement observed on the beam under stress. The main advantage of digital image processing is the calculation of the whole displacement field without physical contact, using a simple low cost camera. This way, a complete displacement map is obtained resulting on a more complete and more accurate analysis than on the traditional approach where only a few displacement points are measured. Two photo sequences were taken during the load tests on a concrete beam and on a Plexiglass bar. The images were subjected to three different algorithms: simple and efficient search (SES), adaptive rood pattern search (ARPS) and the rood pattern particle swarm optimization (RP-PSO). The data obtained from these algorithms was compared with the data from LVDT sensors. As we show, the image processing measurements have an accuracy at least equivalent to the LVDT sensors but with much less equipment and setup requirements.

Keywords: image processing, correlation, displacement measurement, block motion estimation, adaptive rood pattern search and particle swarm optimization.

1 Introduction

Concrete structures are used all over the world. These structures have normally a long life but it is necessary to study how materials change over time, the importance of the reinforcement materials and the aging of the material submitting it to cyclic stress tests. Civil engineering laboratory experiments traditionally use electrical gauges to measure beams displacements. However, these sensors require a complex setup and calibration procedure with risks for their integrity when the beams break.

Due to their cost and installation difficulties, only a limited number of points are usually measured limiting the analysis to local displacements.

Since the early 1980's, several works on digital image correlation techniques have been under development to obtain an accurate knowledge of the displacement field [1-3]. Using a simple consumer level camera and a computer it is possible to take photos during and after the loading of the concrete beams at regular time intervals that document the bending of the structure. After collecting the time series images we can detect the displacements between consecutive images using a mathematical correlation algorithm and extract the information about the material under study. Using this methodology the number of measurement points can be decided after the tests without repeating them. If the image processing algorithm is fast enough, it is also possible to achieve real-time analysis with data extracted during the experiment.

The beam deflection over the image time series is measured following small areas containing random patterns called blocks. Block motion algorithms divide an image in small blocks and follow the blocks under study along the image series matching each block on consecutive images. The capability of the block matching algorithms to correctly identify each block on the next image of the series greatly depends on its pattern. The singularity of each block on its neighbourhoods is an important factor for reducing the block tracking error.

The usefulness of the natural pattern depends on the material it is made of. However, it is possible to create an adequate pattern applying randomly speckled black ink over a white background layer. The size and density of the speckle, the camera resolution and the interval between photos are determinant factors of the resulting accuracy.

When displacement fields are measured using physical sensors, the number of measured points is an issue because the increase in the hardware, the time to get the setup ready and the correspondent cost. However, using image analysis techniques we can have a high density of measured points without a significant increase on the price and complexity of the system. As an example, a trivial image of 512 by 512 pixels can be used to obtain a continuous information field with more than 100 analysis points.

In [4, 5] the results of the traditional approach were compared with the results obtained from digital image processing techniques. The simple and efficient search (SES) and adaptive rood pattern search (ARPS) algorithms were used in these studies to compare the results from the image processing algorithms with the obtained from physical sensors.

In this paper three different image processing algorithms are compared in terms of accuracy and computation time using real images captured during two different laboratory essays. For this study we used real images of laboratory tests because, due to the materials composition, there are non-uniform deformations that must be correctly analysed by the image processing algorithms. Some authors use a virtual sequence of images obtained by interpolation [6, 7] ignoring some particularities of real applications.

2 Review of related work

Carolin et al.[8] presented a non-touching strain measurement method that covers a pre-defined area. In this study, a speckle pattern correlation was used and a photo was taken every 30kN until rupture. The picture was divided into sub-pictures of 128x128 pixels and then a threshold was applied. In each of the blocks the centre of gravity was calculated. With this methodology and based on the speckle correlation they could discover the same sub-picture in the second loading condition. They found their method was very dependent on the resolution of the camera, the area studied and also the pattern.

The instructions described in the above paper were followed for our initial laboratory tests. We also found that edge techniques are very dependent on light conditions. Although the results obtained were acceptable, we had to adapt the level of threshold several times for our photos making this a less efficient system.

Almeida et al. in [4] presented their results obtained with a monitored reinforced concrete beam. They took photos every 30s and compared the results from the image processing with the results from the sensor. Instead of black and white image techniques they applied a block motion algorithm. Also in [5] they compared the results from image processing analysis and the adaptive rood pattern algorithm.

Kurtz et al. [9] used digital image correlation (DIC) in order to obtain information from a load test. Before loading, a random black and white speckle pattern was painted on the surface of the specimen and an unloaded image was taken. At each load stage additional images were acquired. The images were divided into sub images of grey level data. Each sub image was then compared with the next image using the cross-correlation error function. This process was repeated for all sub images in the area of interest, resulting in a full-field map of the surface displacements. They used two different groups of beams, one for the DIC tests and another for the strain-gauged tests. The obtained results are compared with the analytical models.

One of the first issues related to DIC is to choose the best characterization of the speckle pattern and for that particular speckle pattern to decide what is the subset size to be used. In [7, 10] it is possible to see the influence of the speckle size on displacement accuracy. The authors found that the larger the subset, the more accurate the measured displacements are and where a small subset is used, the most precise result is obtained with the smallest speckles.

Another paper related to block size issue is [11]. Here they apply the digital image correlation to polymeric materials using the same method related in [6]. In this paper they compared the results obtained from digital image correlation and the results from the sensors.

3 Block motion algorithms

Digital image correlation examines consecutive images and detects the displacements of the objects present on the images based on a mathematical algorithm. The technique presented in this paper is based on block motion estimation. The basic idea is to divide the current image into a matrix of blocks of NxN pixels and then compare these blocks with the previous image in order to calculate the movement vectors (MV). The current block is searched in the previous image in a delimited search area, p pixels around the current block.

The most commonly used functions to compare the blocks are (1) the mean absolute difference (MAD), (2) the mean square error (MSE) and (3) the cross correlation (CC).

$$MAD = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N}\left|C_{ij} - R_{ij}\right| \tag{1}$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N}\left(C_{ij} - R_{ij}\right)^2 \tag{2}$$

$$CC = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N}\left(C_{ij} - \bar{C}\right)\left(R_{ij} - \bar{R}\right)}{\sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N}\left(C_{ij} - \bar{C}\right)^2 \sum_{i=1}^{N}\sum_{j=1}^{N}\left(R_{ij} - \bar{R}\right)^2}} \tag{3}$$

where N is the side of the block and C_{ij} and R_{ij} are the pixels in the current and reference blocks, and $\bar{C}$ and $\bar{R}$ are the mean of C and R respectively.

In this paragraph the block motion algorithms used in this work are presented.

3.1 Simple and efficient search

The simple and efficient search algorithm (SES) presented by [12] is a variation of the classical three-step search algorithm (TSS). The SES algorithm has two phases in order to reduce computation time taking advantage of the uniformity of the pattern.

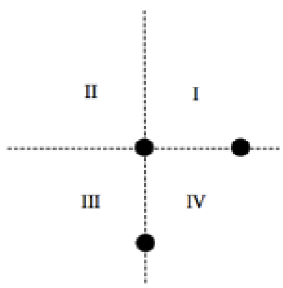

Figure 1. Selection of the search direction quadrant

In each phase the number of steps of the algorithm is dependent on the search window. The first phase consists of selecting the search direction quadrant (see Figure 1). To choose the search quadrant it is necessary to compute the cost function (CF) error for three locations: A, B and C. The most promising quadrant is chosen using the following four rules:

If $CF(A) \geq CF(B)$ and $CF(A) \geq CF(C)$ quadrant I is selected;
If $CF(A) \geq CF(B)$ and $CF(A) < CF(C)$ quadrant II is selected;
If $CF(A) < CF(B)$ and $CF(A) < CF(C)$ quadrant III is selected;
If $CF(A) < CF(B)$ and $CF(A) \geq CF(C)$ quadrant IV is selected.

After selecting the most promising quadrant it is necessary to calculate other points in the chosen quadrant that will help to decide the best block match. This is the second phase. In Figure 2 the initial search patterns of phase 1 (black dots) and phase 2 (white squares) are indicated.

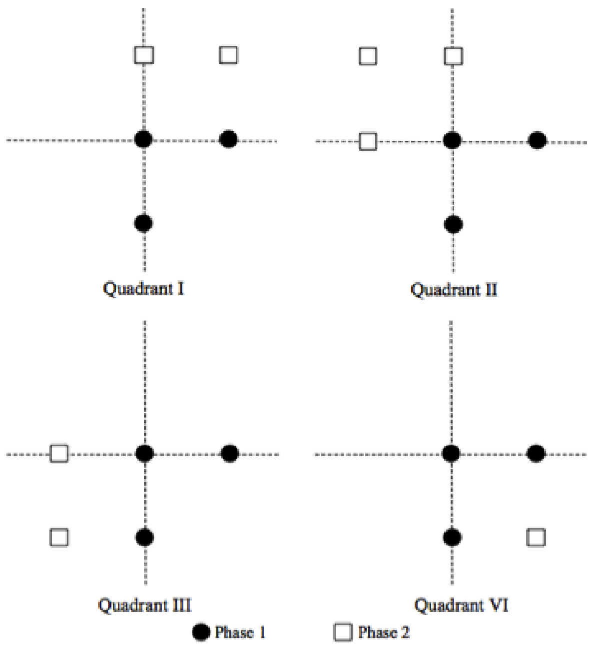

Figure 2. Search patterns for each select quadrant achieved in phase 1

The goal of the second phase is to find the location with the smallest error within the previously selected quadrant. This process is reiterated with the points A, B and C inside the new search window until a null displacement is found.

3.2 Adaptive rood pattern search

The adaptive rood pattern search algorithm implemented in this work was based on the proposal by Nie in [13]. In our specific application the adjacent blocks have similar motions. The blocks on the immediate left, above, above-left and above right of the current block are the most important for calculating the predicted motion vector (MV). Four types of region of support (ROS) according to can be used (Figure 3).

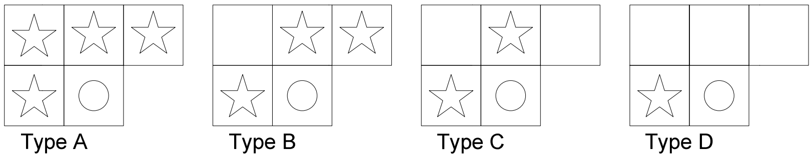

Figure 3. Regions of support: in each case the blocks market with an "O" are the current block and the ones with a star are used for prediction of the motion vector

In this paper the predicted motion vector is based on the immediate left of the current block, type D, because it requires less memory and computation time.

For the initial search, the ARPS algorithm evaluates the four endpoints in a symmetrical rood pattern plus the predicted motion vector (Figure 4). It is therefore necessary to compute the cost function in each search point.

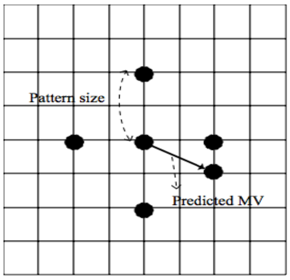

Figure 4. Symmetrical shape of Adaptive Rood Pattern with four search points located at the four vertices

The four arms of the rood pattern are of equal length. The size of the rood pattern is approximately the length of the predicted motion vector (i.e. the motion vector of the immediate left of the current block). The size of the rood pattern, Γ, is calculated in (4).

$$\Gamma = round\left|M\overline{V}_{predicted}\right| = round\left[\sqrt{MV^2_{predicted(x)} + MV^2_{predicted(y)}}\right] \tag{4}$$

The square and the root square operations drawn in (4) require a significant computational effort. Therefore, instead of (4), it is possible to use a simplification that only requires the highest magnitude of the two components of the predicted MV (5).

$$\Gamma = MAX\left\{\left|MV_{predicted(x)}\right|, \left|MV_{predicted(y)}\right|\right\} \tag{5}$$

If it is not possible to apply the type D of the ROS (see Figure 3), the value 2 is chosen for the size of the arm length (i.e. Γ=2). The minimal matching error (MME) point found in the current step will be the starting search centre for the next iteration. The algorithm repeats itself until the MME coincides with the centre of the rood pattern.

3.3 Rood Pattern - Particle Swarm optimization (RP-PSO)

Particle swarm optimization (PSO) is a population based stochastic optimization algorithm designed and proposed by Kennedy and Eberhart [14]. This adaptive algorithm is based on the simulation of the social behaviour of a group of ants, flocking birds or schooling fish.

PSO is a simple algorithm that is effective for optimizing a wide range of functions using basic mathematical operators computationally inexpensive in terms of memory and speed.

In the PSO algorithm a particle is represented in a multidimensional search space The i^{th} particle is represented as $X_I = (x_{i1}, x_{i2}, \ldots, x_{id})$. The algorithm records the best previous position of any particle, $P_I = (p_{i1}, p_{i2}, \ldots, p_{id})$ that is achieved by the best fitness value function. The index of the best particle among all the particles in the population is represented by the symbol *g*. The rate of the position change (velocity) for particle *i* is represented as $V_I = (v_{i1}, v_{i2}, \ldots, v_{id})$.

The position of a particle is influenced by the best position it visited and the position of the best particle in its neighbourhood. The most common neighbourhood topologies are *star* and *ring* topologies. The choice of neighbourhood topology has a profound effect on the propagation of the best solution found by the swarm [15].

A new version of PSO was proposed in [16] by adding a new inertia weight into the original algorithm designed by [14]. Also, the PSO algorithm has been used for solving the block matching problem by [17] [18] and [19].

Bakward et al [17] used the PSO for motion estimation in a video sequence. They claimed that their method saves computational time by up to 94%. In that paper they also compared their method with SES and ARPS among other block motion algorithms. Also related to a video sequence, Ranganadham et al. [18] proposed a methodology based on PSO algorithm to calculate the bidirectional motion. These two papers inspired us to combined them and present a variation with PSO and ARPS algorithms.

Generally, in block matching algorithms, a swarm is divided into overlapping neighbourhoods of particles. For each neighbourhood the best particle is determined using the minimum cost function. This particle is referred to as the neighbourhood best particle, *lbest*. In parallel particle swarm optimization [20], *lbest* is updated based on the fitness function value of all swarms, while in serial PSO it is updated from the fitness function value of individual swarms.

The particles evaluation is done according to the following equations:

$$v_{id} = W * v_{id} + C_1 * R1 * (p_{id} - x_{id}) + C_2 * R2 * (p_{gd} - x_{id}) \tag{6}$$

$$x_{id} = x_{id} + v_{id} \tag{7}$$

where c_1 and c_2 are two positive constants, *R1* and *R2* are two random functions in the range [0,1], and *W* is the inertia weight.
Our intention was to combine the advantages of the rood pattern search with the PSO algorithm and apply this new algorithm, called rood pattern-particle swarm optimization (RP-PSO), to the load tests instead of a video sequence. We used 8 particles around the point where we want to calculate the motion vector: four particles are positioned close to the central point and the other four particles are initially forwards (see Figure 5). The motion vector of the left block is used as the initial velocity or, if this information is not available because we are in image boundaries, the number zero is used. In order to avoid going beyond the boundaries of an image a maximum velocity should be chosen to truncate the velocity vector. With this organization it is possible to cover a larger area in all directions.

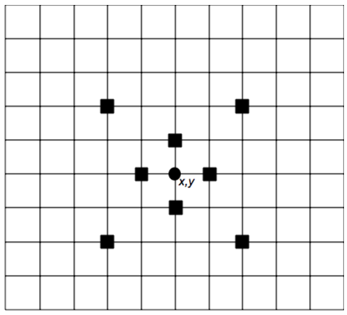

Figure 5. The particle (x,y) represents the centre of the current block. The black squares represent the centre of the neighbours

For each neighbourhood the best particle is determined. This particle is referred to as the *local best* particle (*lbest*). With this information all the neighbourhood particles are updated in one step using the fitness function value of all swarms. Using only local information, p_l, the equations (6) and (7) are simplified:

$$v_{id} = W * v_{id} + C_1 * R * (p_{ld} - x_{id}) \quad (8)$$

$$x_{id} = x_{id} + v_{id} \quad (9)$$

The implemented block motion algorithm based on RP-PSO can be summarized as:

1. Initialize the position of the swarm's particle, the rood pattern plus the initial velocity.
2. Initialize the position of the swarm's particle according to equation (9);
3. Evaluate the fitness function for each particle, using equation (1), (2) or (3);
4. For each particle compare the particles' fitness value and identify the particle that has the best fitness value and its position;
5. Update the velocities and positions for all particles using equations (8) and (9); if the velocity vector is greater than maximum than is necessary to truncate to the maximum allowed;
6. Repeat steps 3 to 5 until a fixed number of iterations is achieved.

4 Experimental Results

Several load tests have and are being carried out in partnership with researchers from the Civil engineering Department of the Universidade Nova de Lisboa (UNL). The data obtained with a concrete beam and a Plexiglass bar was used in this paper. In this section the results of our experimental tests are presented.

4.1 Algorithmic Parameters

In all algorithms the MAD function was used as the fitness function. Only the information of the intensity of the pixel was used as a feature. Block sizes of 32x32, 64x64, 128x128 and 256x256 were used in the tests.

In the SES and ARPS algorithms the value 15 was used for the search parameter p. In the RP-PSO algorithm two iterations were done and the maximum velocity vector allowed was 15, C1 was 2 and the inertia weight W was 0.3. The image acquisition conditions are shown in Table 1.

T beam designation	Resolution [Pixel/cm]	Interval between images [s]	Number of images
Plexiglass	196	5	32
T++10000_05	84	10	42

Table 1. T-Beam image series acquisition conditions

The beam was prepared with an underlying layer of white matt ink on top of which a superimposed random speckle pattern was manually applied using a large brush with black matt ink. The random speckle pattern in the Plexiglass was manually sprayed with black matt ink.

The deflection measurement was supported by standard 100 mm linear voltage displacement transducer sensors (LVDT) placed along the longitudinal direction of the beam. The data from the LVDT sensor at the mid-span was used for the comparison with the data obtained from the image analysis. In order to obtain a ground truth, three trained users that classified for three times each image series manually evaluated images.

The image acquisition was done using a digital Cannon EOS 400D camera with a resolution of 3888x2592 and two spotlights of 500W. Figure 6 show the random pattern applied to the bars.

Table 2 shows the computation time spent by each algorithm. The RP-PSO is in middle of SES and ARPS algorithm in terms of computation time but as we will see later RP-PSO achieves better measurements than ARPS algorithm.

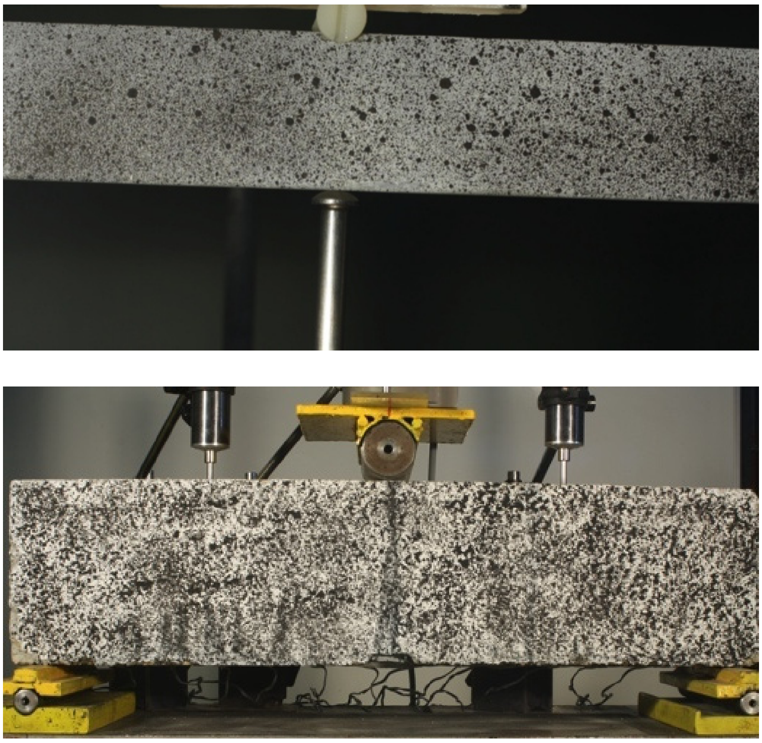

Figure 6. Example of the random pattern applied to the Plexiglass (at top) and T++10000_05 (at bottom)

T beam designation	Number of Blocks	SES per image [s]	ARPS per image [s]	RP-PSO per Image [s]
Plexiglass	91	0,3243	0,2502	0,2868
T++10000_05	640	1,6259	1,1308	1.3996

Table 2. Computation time used by the different algorithms with a 64x64 block size

The results obtained with the LVDT sensor were compared with those obtained at the middle of the beams.

In all the comparative figures, the yellow spots represent the mean of our ground truth that is represented on a box plot.

Figure 7 shows the results for the Plexiglass with block sizes of 32x32, 64x64 and 128x128 pixels. The poor results obtained with the 32x32 block size show that this size insufficient for our experimental conditions. With blocks of 64x64 pixels the SES algorithm is better than RP-PSO but its computation time is the worst. The ARPS algorithm diverges from our ground truth in the second part of the experiment. All the three algorithms obtain the best results with blocks of 128x128 pixels.

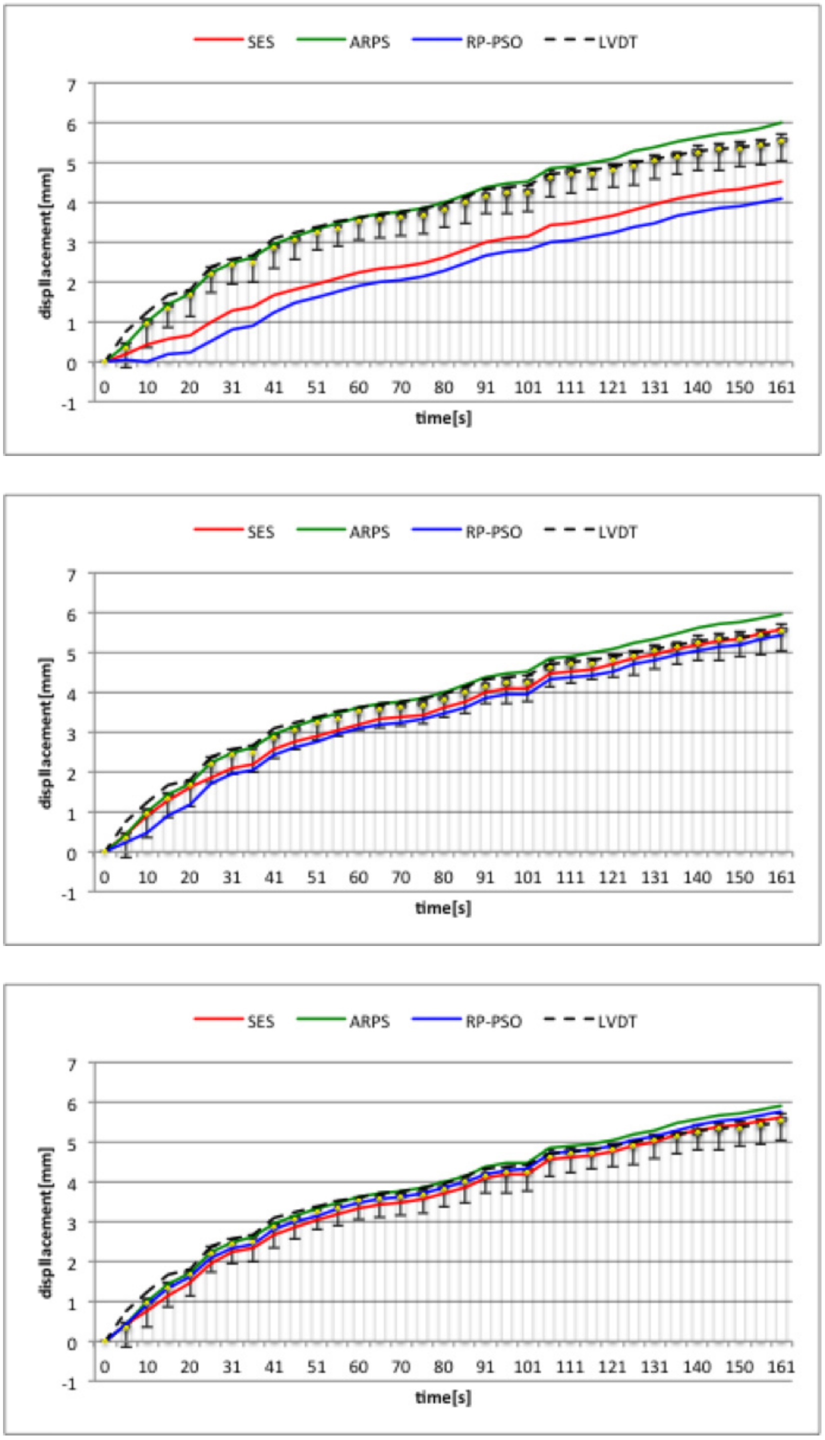

Figure 7. Plexiglass ground truth compared with LVDT and image processing algorithms with a block size of 32x32, 64x64 and 128x128 pixels respectively

With the concrete beam, T++10000_05 blocks of 64x64, 128x128 and 256x256 were used (Figure 8).

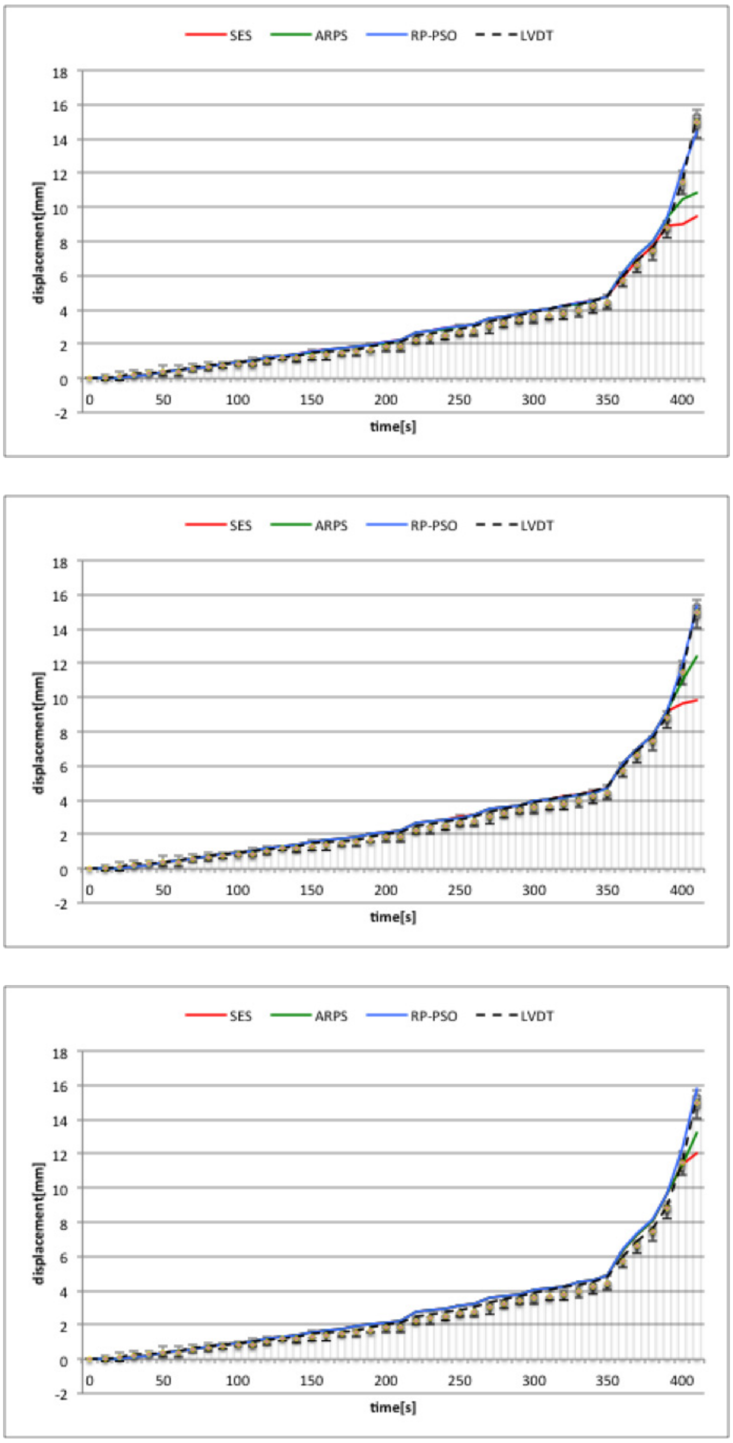

Figure 8. T++10000_05 ground truth compared with LVDT and image processing algorithms with block sizes of 64x64, 128x128 and 1256x256 pixels respectively

As it is visible in the previous figure, the SES and ARPS algorithms improve their results with the increase on the block size. Despite that all the algorithms show some difficult to follow abrupt displacements that happen when cracks occur, their ability to adapt to this circumstances is enhanced by the increase on the block as it can be seen on the last part of the experiments shown in Figure 8. The RP-PSO shows better dynamics for all block sizes.

When compared with our ground truth the RP-PSO mean error for Plexiglass is 5% and for T++10000 is 3,2%.

The total displacements in the y-axis for the complete images time series are presented in Figure 9. Red regions identify the points with larger displacement predicting the appearance of cracks that happen later in the experiment. It is possible to see that the SES algorithm is the more sensitive.

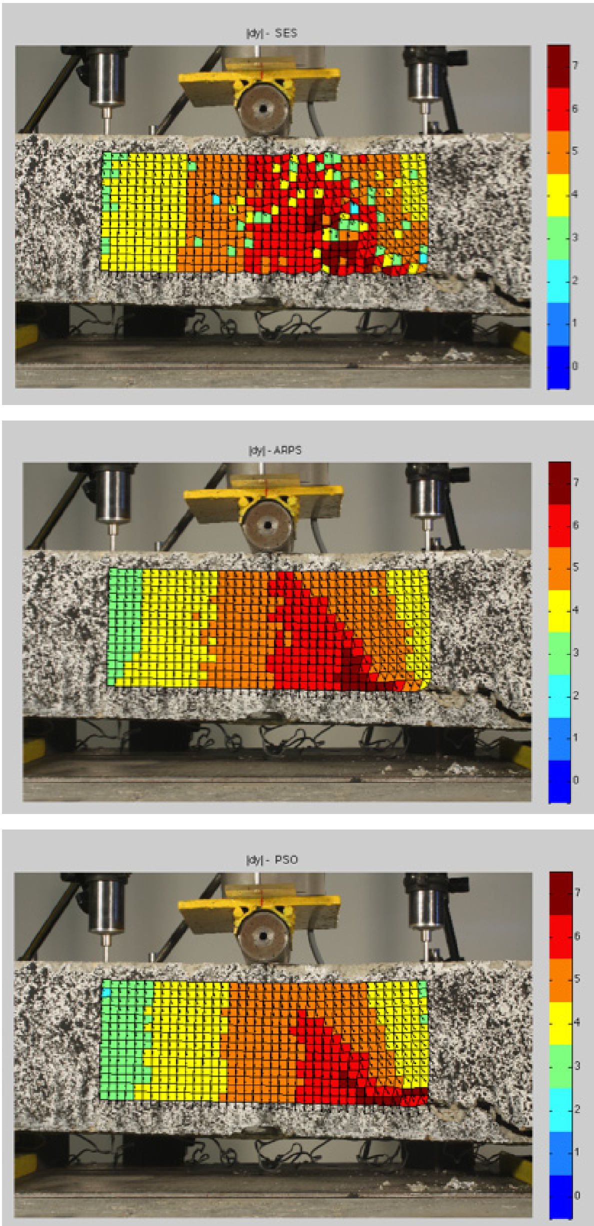

Figure 9. Displacement in the y-axis for the T++10000_05 using SES, ARPS and RP-PSO with block size of 128x128 pixels

5 Conclusions

In this paper we used three different image processing algorithms for the measurement of displacement measurements in civil engineering load tests. To test the algorithms two experiences were carried out using different beam materials: concrete and Plexiglass. The random pattern superimposed on the Plexiglass revealed easier to analyse and more appropriate for image processing due to its smaller size.

All the algorithms shown acceptable accuracy results when compared with the LVDT sensors and with the ground truth obtained from three different trained users. The new algorithm proposed (RP-PSO), a mix between ARPS and PSO, shows less dependency on the block size used on the image analysis and a higher dynamics that facilitates the tracking of the blocks when abrupt displacements occurs. The inconvenience of RP-PSO is a slightly higher computational time that is compensated by the best results achieved. The efficiency of all the algorithms is dependent on several factors such as the speckle pattern, the block size, the search parameters, the image resolution and the time between images. However, as we show in this study, it is not difficult to find a good compromise on these values that enables good results. The comparison of the image processing algorithms with the ground truth obtained from the trained users is relevant since the LVDT sensors also have a considerable error.

The main advantages of the image processing techniques are the easy to setup and inexpensive equipment required and the production of a displacement map that shows how the material reacts to the stress that is imposed during the experiment. Due to the unpredictable behaviour of the beam structure, the large area covered by the images also guarantees the analyses of the all beam that is extremely difficult or even impossible to do with traditional sensors. Moreover, the image documentation also allows the detailed analysis of small areas of interest with greater detail even after the experiment is complete.

The automatic analysis of the images time series is an important factor for the adoption of this technology in this area. The proposed RP-PSO algorithm demonstrates it to be adequate for this proposal showing advantages over the alternative approaches.

References

[1] M. A. Sutton, W. J. Wolters, W. H. Peters, W. F. Ranson, and S. R. McNeill, "Determination of displacements using an improved digital correlation method," presented at the Image and Vision Computing, 1983.

[2] M. Sutton, J. Turner, H. Bruck, and T. Chae, "Full-field representation of discretely sampled surface deformation for displacement and strain analysis," *Experimental Mechanics,* vol. 31, pp. 168-177, 1991.

[3] M. A. Sutton, J.-J. Orteu, and H. Schreier, *Image Correlation for Shape, Motion and Deformation Measurements*: Springer, 2009.

[4] G. Almeida, H. Biscaia, F. Melicio, C. Chastre, and J. Fonseca, "Displacement Estimation of a RC Beam Test based on TSS algorithm," presented at the Information Systems and Technologies (CISTI), 2010 5th Iberian Conference on Santiago de Compostela, 2010.

[5] G. Almeida, F. Melicio, and J. Fonseca, "Displacement measurements with block motion algorithms," presented at the VIPIMAGE, III ECCOMAS THEMATIC CONFERENCE ON COMPUTATIONAL VISION AND MEDICAL PROCESSING, Olhão, Portugal, 2011.

[6] B. Pan, H. Xie, Z. Wang, K. Qian, and Z. Wang, "Study on subset size selection in digital image correlation for speckle patterns," in *Optics Express*, 2008

[7] D. Lecompte, H. Sol, J. Vantomme, and A. Habraken, "Analysis of speckle patterns for deformation measurements by digital image correlation," presented at the SPIE, Nimes,France, 2006.

[8] A. Carolin, T. Olofsson, and B. Taljsten, "Photographic strain monitoring for civil engineering," *FRP Composites in Civil Engineering - CICE 2004, Seracino,* 2004.

[9] S. Kurtz, P. Balaguru, and J. Helm, "Experimental Study of Interfacial Shear Stresses in FRP-Strengthened RC Beams," *Journal of Composites for Construction,* vol. 12, p. 10, 2008.

[10] P. L. Reu, M. Sutton, Y. Wang, and T. J. Miller, "Uncertainty quantification for digital image correlation," presented at the Proceedings of the SEM Annual Conference, 2009 Albuquerque New Mexico USA, 2009.

[11] M. Jerabek, Z. Major, and R. W. Lang, "Strain determination of polymeric materials using digital image correlation," *Polymer Testing,* vol. 29, pp. 407-416, 2010.

[12] L. Jianhua and M. L. Liou, "A simple and efficient search algorithm for block-matching motion estimation," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 7, pp. 429-433, 1997.

[13] Y. Nie and M. Kai-Kuang, "Adaptive Rood Pattern Search for Fast Block-Matching Motion Estimation," *IEEE TRANSACTIONS ON IMAGE PROCESSING,* vol. 11, p. 8, 2002.

[14] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *IEEE International conference on Neural Networks*, Perth, WA, Australia, 1995, pp. 1942-1948.

[15] M. G. H. Omran, "Particle Swarm Optimization for Pattern Recognition and Image Processing," University of Pretoria, Pretoria, 2004.

[16] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, 1998, pp. 69-73.

[17] K. M. Bakwad, S. S. Pattnaik, B. S. Sohi, S. Devi, S. Gollapudi, C. V. Sagar, and P. K. Patra, "Small Population Based Modified Parallel Particle Swarm Optimization for Motion Estimation," in *Advanced Computing and Communications, 2008. ADCOM 2008. 16th International Conference on* Chennai, 2008, pp. 367-373.

[18] D. Ranganadham, P. Gorpuni, and G. Panda, "An Efficient Bidirectional Frame Prediction Using Particle Swarm Optimization Technique," in *Advances in Recent Technologies in Communication and Computing, 2009. ARTCom '09. International Conference on*, 2009, pp. 42-46.
[19] X. Yuan and S. Xiaojing, "Block Matching Algorithm Based on Particle Swarm Optimization for Motion Estimation," in *Embedded Software and Systems, 2008. ICESS '08. International Conference on*, 2008, pp. 191-195.
[20] J. F. Schutte, J. A. Reinbolt, B. J. Fregly, R. T. Haftka, and A. D. George, "Parallel global optimazation with the particle swarm algorithm," *International Journal for Numerical Methods in engineering,* vol. 61, pp. 2296-2315, 2004.