

Article

Hardware Implementation for an Improved Full-Pixel Search Algorithm Based on Normalized Cross Correlation Method

Guohe Zhang¹, Zejie Kuang¹, Sufen Wei^{1,3}, Kai Huang¹, Feng Liang¹ and Cheng-Fu Yang^{2,*}

- ¹ School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China; zhangguohe@xjtu.edu.cn (G.Z.); kuangzejie@stu.xjtu.edu.cn (Z.K.); weisufen@jmu.edu.cn (S.W.); huangkailnn@126.com (K.H.); fengliang@xjtu.edu.cn (F.L.)
- ² Department of Chemical and Materials Engineering, National University of Kaohsiung, No. 700, Kaohsiung University Rd., Nan-Tzu District, Kaohsiung 811, Taiwan
- ³ School of Information and Engineering, Jimei University, Fujian 361021, China
- * Correspondence: cfyang@nuk.edu.tw; Tel.: +886-91-7270840

Received: 7 November 2018; Accepted: 8 December 2018; Published: 12 December 2018



Abstract: Digital speckle correlation method is widely used in the areas of three-dimensional deformation and morphology measurement. It has the advantages of non-contact, high precision, and strong stability. However, it is very complex to be carried out with low speed software implementation. Here, an improved full pixel search algorithm based on the normalized cross correlation (NCC) method considering hardware implementation is proposed. According to the field programmable gate array (FPGA) simulation results, the speed of hardware design proposed in this paper is 2000 faster than that of software in single point matching, and 600 times faster than software in multi-point matching. The speed of the presented algorithm shows an increasing trend with the increase of the template size when performing multipoint matching.

Keywords: full pixel search algorithm; digital speckle correlation measurement method; hardware implementation

1. Introduction

With the rapid development of advanced manufacturing technology, the research of three-dimensional measurement technology is widely used in the fields of material testing, strength testing, and quality control. Digital speckle correlation measurement is an important method in the field of modern optical measurement [1]. Compared with other optical measurement methods, the most significant advantage lies in its simple experimental equipment and convenient measurement. It has the characteristics of full field non-contact, no damage, high accuracy, and high automation by using binocular image acquisition equipment [2–4]. Because of these characteristics, the digital speckle correlation method breaks the limitations of traditional measurement methods and creates a wide range of applications, such as the measurement of deformation of composite materials [5], the detection of cancer cells by measuring the skin strain [6], and measurement in a complex and severe environment [7–10], et al.

The basic idea of digital speckle correlation method is to divide the search process into two parts, the full-pixel search and the sub-pixel search [11]. The full-pixel search is performed in integer pixels, and rapidly locates in a large area. The sub-pixel search is done on the basis of full-pixel to further improves the accuracy. Most applications require a precision in the order of 1% of a pixel, and only the sub-pixel search can achieve this accuracy requirement. Obviously, the full-pixel search is very essential to the feasibility, accuracy, and high speed of subsequent sub-pixel search. The low speed during the



matching search is the main problem needed to be solved especially in the case of high computational complexity and large amount of data processing [12]. Normalized cross correlation (NCC) method based on gray features is used as the similarity measure function in the image matching search [13]. It has the characteristics of high accuracy, good performance such as anti-noise and adaptability to image distortion. However, problems such as complex correlation computation, low computation speed, and long computation time still exist with software implementation. The limitations are unsuitable for some real-time system applications. Nowadays, the hardware implementation of image algorithms has been widely applied in the field of image processing with the advantages of high degree of parallelism, integration and resulting high speed, low power consumption, and low cost [14,15]. The implementation of image algorithms through hardware can overcome the computational bottleneck and improve the processing speed of the system.

Here, an improved fast full-pixel search algorithm considering hardware implementation is proposed to solve the existing problems of low speed and high complexity in the NCC algorithm. A compromise strategy of rough matching and exact matching is adopted. When the system is implemented in FPGA (Field Programmable Gate Array), the corresponding number of multipliers is used to calculate the correlation coefficient of the local matching template and the time-sharing method is adopted to speed up the full template calculation.

2. Improvement and Hardware Design of Full-Pixel Search Method

2.1. Digital Speckle Correlation Calculation and Correlation Coefficient

The digital speckle correlation matching is to obtain the displacement and the deformation of an object according to the cross correlation of the speckle images before and after the object is deformed. The digital speckle correlation calculation process is shown in Figure 1.



Figure 1. Calculation procedure of Digital Speckle Correlation Method.

The gray scale distribution function of a reference image before deformation is set to f(x, y). The distribution function of a target image after deformation is set to g(x, y). In the reference picture, a rectangular area of size $N \times N$ centered on a point P(x, y) to be measured is selected as a sample sub-area which usually become a matching template. At the same time, in the target picture, a rectangular area of size $M \times M$ (M is greater than N) centered on P(x, y) is selected as a search sub-area which usually become a matching window. Then correlation operations are performed in the search sub-area with the sub-area B, which is of the same size as the sample sub-area, to find the point Q(x, y) corresponding to the extreme value (dependent on the correlation function) of the correlation coefficient of the sample sub-area selected in the reference picture. Thereby, the displacement component of the reference point P(x, y) in the *x* and *y* directions can be determined.

The digital speckle correlation method generally uses a gray-based statistical correlation algorithm. Its principle is to determine whether the two sub-regions are related according to the statistical characteristics of the gray distribution of each sub-region before and after the deformation. Usually, the judgment is based on various correlation coefficient functions. This paper adopts normalized cross correlation coefficient [16,17].

2.2. Improved Fast Integer Pixel Search Method

The flow chart of the overall implementation of the algorithm is shown in Figure 2. Firstly, the reference image and the target image are imported. The reference sub-area is selected as the full template of the match in the reference image, and the initial search area is selected in the target image. Then, the local template slides from left to right and from top to bottom in the search area. Correlation function is used to calculate the correlation coefficient value in each sliding process and to compare the calculated correlation coefficient value with the threshold. If the correlation coefficient value is greater than the threshold value, it means that the search window has strong correlation with the reference sub-area and can be used as a candidate matching window. If the correlation coefficient value is less than the threshold, the correlation coefficient value of the next window will be continued to calculate. At the same time, the histogram statistics strategy is used to count the correlation coefficient values in the whole local template matching process. Then a full template is used to calculate the correlation coefficient values of all candidate matching windows to determine the best matching point, and the search task for a pair of matching points is completed. Finally, the correlation coefficient values of the histogram statistics are processed to calculate the optimal threshold of the next pair of matching points. The position and size of the search area of the next matching point are adjusted adaptively by the displacement component of the matching point.



Figure 2. Overall algorithm flow chart.

The improved algorithm mainly includes the following three key steps:

(1) Selection method of local matching templates

The two-layer matching algorithm is adopted in this paper. Compared with the three-layer matching algorithm [18], the multiple matching tedious steps are reduced. As an important step to improve the calculation efficiency of the algorithm, local template matching not only requires as few pixels as possible, but also needs to reflect the texture information of the full template as much as possible. Therefore, regional blocks are selected at various locations in the full template to form a local matching template. As shown in Figure 3, five regions R0, R1, R2, R3, and R4 were selected to form a local template. R0 is distributed in the middle of the full template which is made up of rectangular areas of $M \times M$ size, and the other four regions are distributed on the four corners of the full template. These five regions are distributed in different positions of the template and can effectively represent the texture information of the entire template. If the size of the local template is larger, the accuracy of the matching is higher, but correspondingly, the consumption of resources is greater. When assuming the entire template as a rectangular area of size $H \times H$, a rectangular region of $h \approx H/4$ can be selected as a local matching template considering the accuracy and resource consumption. If a 31×31 full template is used, the edge length of the local template can be selected as 7 pixels. The purpose of the local template method is to exclude some non-matching points and thus reduce the computational complexity. The best match point is obtained after the full template calculation.



Figure 3. Template for local matching.

(2) Adaptive selection method of thresholds for histogram statistics

We use the correlation coefficient of the local template as the threshold. The closer the correlation coefficient is to 1, the stronger the correlation between the two sub-areas (reference sub-area and search sub-area) is; so is the threshold. The choice of threshold is extremely important to the whole algorithm. The correlation coefficient calculated by the local template needs to be compared with the threshold to exclude some poorly correlated points. If the selected threshold is too small, a large number of matching points will be retained, and the total number of full template calculations will be increased. If the selected threshold is too large, the best matching point might be eliminated which will be resulting in a greater likelihood of image mismatch.

In order to improve the reliability and flexibility of the algorithm, an adaptive threshold selection method is presented. Image correlation matching algorithm usually needs to perform ergodic correlation operations on all pixels of the speckle image before and after the entire displacement to determine the displacement vector of the whole field pixel (as shown in Figure 4). There is an overlapping area between the reference sub-area and the search area of two adjacent matching points. The texture information, the noise, and the exposure effects of them are similar. Therefore, a certain similarity can be seen in the distribution of their correlation coefficients. Based on this feature, the distribution of correlation coefficients of the current matching points under the local matching template can be collected to obtain the number of correlation coefficient values in each interval. The appropriate correlation coefficient value can be calculated as the threshold of the next matching points. The overlapping search areas of two adjacent points A and B to be matched are shown in Figure 5. We use the histogram to make statistics on the correlation coefficient of matching point A under the local template. The correlation coefficient interval $0 \sim 1$ is subdivided into 100 intervals as the abscissa of the histogram, and the length of each interval is 0.01. The ordinate of the histogram represents the number of matching points in a certain interval. When the local matching template is swiped once in the search area, a correlation coefficient is calculated. The interval is determined by where the correlation coefficient is located. The histogram of the corresponding interval plus 1 is generated to complete the statistics of the correlation coefficient. When the matching point search is completed, the number of occurrences of the correlation coefficient in each interval can be obtained. After completing the statistics, the number of occurrences is sequentially accumulated from the larger interval to the smaller interval of the histogram. So, accumulation stops when the accumulated value reaches 8% of the total amount, and the current abscissa is used as the search threshold of matching point B. Because the full template only needs to participate in the calculation of about 8% window in the search area, a large amount of non-matching points can be excluded by this method. It greatly reduces the amount of calculations, saves search matching time, and improves the matching efficiency of the algorithm.



Figure 4. Full field whole pixel displacement vector.



Figure 5. Overlapping regions of adjacent matching points.

(3) Adaptive adjustment of the location of the search areas

When industrial measurements are made using digital speckle, the displacement or deformation of each point on the surface of a object can be considered to vary continuously. In general, sudden mutations are less likely to occur, so it can be considered that the displacement values of adjacent two matching points in the speckle pattern before and after the displacement are not significantly different. In the traditional full-pixel search algorithm, a rectangular area which is centered on the coordinates of the center point of the reference sub-region and larger than the reference sub-area is framed in the target image as the search area. Therefore, in the case of uncertain object displacement, it is necessary to select a larger search area to prevent the best matching point from image mismatch. However, if the search area is too large, the number of searches increases and the search speed decreases.

In order to solve the above problems, a method for adaptively adjusting the search area is presented, as shown in Figure 6. Assuming the point *P* in the first picture is the initial point to be matched of the reference image, a larger rectangular area is selected as the search area *P* centered on the coordinates of the point *P* in the target image, as shown in the second picture. When the search is completed, the best matching point P^* is obtained, and the displacement (u, v) of the current matching point is recorded. When searching for the adjacent matching point *Q*, *Q*^{*} corresponding to point *Q* after the displacement (u, v) is found in the target image. Then, taking point *Q*^{*} as the center, a rectangular area smaller than the search area *P* is selected as the search area of the *Q* point, as shown in the third picture. The size of the initial search area and the adjusted search area are configurable. In the case of sudden mutations, the size of the adjusted search area to complete the search. The displacement of the point *Q* is recorded, and the next search area is adjusted with it. The method improves the flexibility of the search area selection, further reduces the search time and improves the efficiency of the algorithm matching search by adjusting the position and size of the search area adaptively.



Figure 6. Adaptive adjustment of search areas.

3. Hardware Circuit Design

The full-pixel search matching is a crucial step as well as the most complex step in the digital speckle correlation method. Considering the improvement of resource utilization and the matching speed, a hardware implementation circuit of integer pixel fast matching algorithm is designed.

Through the improvement of the matching search algorithm, the fast matching algorithm proposed in this paper can be better suited to the hardware implementation. The structure of the hardware implementation is shown in Figure 7. The buffer unit reads in the data of the reference image and the target image serially and outputs them parallelly. The algorithm is implemented by using the corresponding number of multipliers to calculate the correlation coefficient of the local matching template firstly. If the full template calculation is required after the threshold comparison, the multiplier array is multiplexed by time-sharing method. It ensures that the full template calculation for this window is completed in three clock cycles. The cost of hardware implementation is greatly reduced.



Figure 7. Structure of hardware implementation.

The hardware implementation adopts the matching template of size 31×31 . The pixel size of the reference image and the search image is 512×512 , the search area size of the initial point is 256×256 , and the size of the search area after the adaptive adjustment is 151×151 .

3.1. Buffer Unit

(1) Data preprocessing module

In this paper, two Synchronous Dynamic Random Access Memories(SDRAMs) are used as storage units for the reference image and the target image. The structure of the SDRAM control unit is shown in Figure 8.



Figure 8. Structure of the SDRAM control unit.

The SDRAM requires a drive clock and control clock. A fixed phase difference between the drive clock and the control clock is required to ensure stable read/write data. Therefore, a PLL is added to the design. The external clock separates two clocks with the same frequency and different phases through the PLL. The sequential control of the SDRAM is composed of the Command module and the Control_interface module. The Command module is primarily responsible for the control of precharge, refresh and burst read/write. Control_interface module controls the operation of SDRAM directly. First, the SDRAM is initialized. Then the SDRAM performs normal read/write operations. All of the rows and columns of Banks must be pre-refreshed every 64 ms to ensure that data is not lost. The read/write dual ports are the SDRAM_WR_FIFO input port and the SDRAM_RD_FIFO output port. Since the read/write operations of the SDRAM share a set of I/O ports, the SDRAM can only perform the read operation or the write operation at the same time. The asynchronous FIFOs are used to achieve cross-clock domain interaction with the data. The control operation of the FIFOs is as follows:

When a write request occurs, the data is sent to the WR_FIFO for caching;

When a read request occurs, the data is sent to the RD_FIFO for caching;

When the data in the WR_FIFO exceeds a certain depth, the SDRAM write operation is requested, and the burst write operation is started when the response is received;

When the data in the RD_FIFO is less than a certain depth, the SDRAM read operation is requested, and the burst read operation is started when the response is received.

(2) Parallel data output module

The parallel data output module reads the data of the reference sub-area and the search sub-area and performs serial-to-parallel processing on the data of the search sub-area. Figure 9 shows the timing of the reference sub-area data output. First, send a command to SDRAM to read the data of the reference sub-area (31 rows, 31 columns) in the burst mode. Ref_Data_En is the enable signal of the row. Every enable single contains 31 valid data. Then, read the data of the next row after a certain time interval. The amount of data in the reference sub-area is significantly less than that in the search area. The read time of the reference sub-area data is shorter than the read time of the search area data. Therefore, the data output to the reference sub-area can be taken in a serial way.



Figure 9. The timing of the reference sub-area data output.

Figure 10 shows the timing of the search sub-area data output. It is similar to the reference sub-area.



Figure 10. The timing of the search sub-area data output.

Then, the data is sent to the buffer unit for serial to parallel processing. As shown in Figure 11. It contains 31 sets of dual port Rams (Dpram) with a data depth of 256 and a data bit width of 8 bits. The Dprams are connected in a serial way. When the high level of the RD_FIFO_Enable signal is detected, the read address of the buffer unit is incremented by 1 and is used as the write address of the next clock cycle. The Search_Data is delayed by one clock and then connected to the Din of the Dpram. When the read address is increased to Size_SR-1, it indicates that all the row data in the Dpram has been read. At this time, clear the read address immediately and continue reading the data.



Figure 11. Dual port Rams (Dprams) of the search sub-area.

The data of a row will be read in one clock cycle. When 31 rows of data are read, the output of the port Dout $1\sim31$ of the buffer unit is the parallel input of the first 31 rows of the search area. The 31 sets of parallel data are valid data for correlation matching operations.

(3) Matching window

The data of the reference sub-area remains unchanged throughout the search process, and its amount of data is small. Once the position of the reference sub-area is determined, the data is immediately read from the SDRAM and stored in the register group. As shown in Figure 12, it is a register group consisting of 31×31 registers. Every register can store the 8-bit data of grayscale image.

Da

$\begin{array}{c} \text{Reg31_31} \\ \text{ta_In} & \text{D} & \text{Q} \\ - & \text{EN} \\ - & \text{> CLK} \end{array}$	$\begin{array}{c} \text{Reg31_30} \\ \hline D & Q \\ \hline - & \text{EN} \\ \hline - & \text{CLK} \end{array}$	$\begin{array}{c} \text{Reg31_29} \\ \hline \text{D} & \text{Q} \\ \hline \text{EN} \\ \hline \text{CLK} \end{array}$	$Reg31_28$ $D Q - \cdots$ $P - EN$ $-> CLK$	$Reg31_4$ $D Q$ EN CLK	$\begin{array}{c} \text{Reg31}_3\\ \hline D & Q\\ \hline \text{EN}\\ \hline \end{array} \\ \hline \end{array} \\ \begin{array}{c} \leftarrow \\ \end{array} \\ \end{array} \\ \begin{array}{c} \leftarrow \\ \end{array} \\ \begin{array}{c} \leftarrow \\ \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \leftarrow \\ \end{array} \\ \end{array} \\ \end{array} \\ \end{array} \\ \begin{array}{c} \leftarrow \\ \end{array} \\ \begin{array}{c} \leftarrow \\ \end{array} \\$	Reg31_2 D Q EN -> CLK	$ \begin{array}{c} \text{Reg31_1} \\ \hline D & Q \\ \hline \bullet & \text{EN} \\ \hline \bullet & \text{CLK} \end{array} $
Reg30_31 D Q EN > CLK	$\begin{array}{c} \text{Reg30_30} \\ \hline D & Q \\ \hline - & \text{EN} \\ \hline - & \text{CLK} \end{array}$	Reg30_29 — D Q — EN — > CLK	Reg30_28 D Q EN -> CLK	$\begin{array}{c} \operatorname{Reg30_4} \\ D & Q \\ \end{array}$	Reg30_3 D Q EN CLK	Reg30_2 — D Q — EN —> CLK	$ \begin{array}{c} $
Reg i_31	Reg i_30	Reg i_29	Reg i 28	Reg i_4	Reg i_3	Reg i_2	Reg i_1
$\begin{array}{c} \text{Reg2}_{31} \\ \text{D} & \text{Q} \\ -\text{EN} \\ -\text{> CLK} \end{array}$	$ \begin{array}{c} \text{Reg2}_{30} \\ \hline \\ D \\ Q \\ \hline \\ \text{EN} \\ \hline \\ \text{-> CLK} \end{array} $	Reg2_29 — D Q — EN — > CLK	Reg2_28 D Q EN -> CLK	$Reg2_4$ $D = Q$ EN $-> CLK$	$ \begin{array}{c} \text{Reg2}_3 \\ \hline D & Q \\ \text{EN} \\ \hline - & \text{CLK} \end{array} $	Reg2_2 D Q EN > CLK	$ \begin{array}{c} Reg2_1 \\ D Q \\ EN \\ -> CLK V $
Reg1_31 D Q EN -> CLK	$Reg1_{30}$ $D Q$ $EN \forall$ CLK	$\begin{array}{c} \text{Regl}_{29} \\ \hline \text{D} & \text{Q} \\ \hline \text{EN} \\ \hline \text{O} & \text{CLK} \end{array}$	$\begin{array}{c} \text{Regl}_{28} \\ \text{D} & \text{Q} \\ \text{-EN} \\ \text{-CLK} \end{array}$	$\begin{array}{c} \text{Regl}_4\\ \text{D} & \text{Q}\\ \text{EN}\\ \text{CLK} \end{array}$	$ \begin{array}{c} \text{Regl}_3 \\ \hline D & Q \\ \hline \bullet & -\text{EN} \\ \hline \bullet & \text{CLK} \end{array} $	Reg1_2 — D Q — EN —> CLK	$ \begin{array}{c} \text{Regl}_1 \\ D & Q \\ \hline \\ \hline \\ \hline \\ \hline \\ \\ \hline \\ \\ \\ \\ \\ \\ \\ \\$

Figure 12. Serial shift register groups of the reference sub-area.

For the sliding window of the search sub-area, the module consists of 31 parallel shift register groups. The hardware implementation is shown in Figure 13. The input of the register group is 31-Channel 8-bit parallel output of the Dpram. The 31×31 parallel shift register group is followed by a First Input First Output buffer(FIFO) and another set of 31×31 parallel shift registers. The result of calculating the correlation coefficient of the local matching template requires a delay of several tens of clock cycles. During these tens of clock cycles, the data of the previous 31×31 parallel shift register groups in the data of the current window is updated early and the full template calculation cannot be performed. Therefore, the FIFO is used to calculate the correlation coefficients of local templates. Register groups in the dotted wireframe are used to calculate correlation coefficients of full templates.



Figure 13. Serial shift register groups of the search sub-area.

The data of the reference sub-area of a matching point is fixed as a matching template throughout the search process. Therefore, each matching point only needs to calculated once.

The circuit structure shown in Figure 14 is the sum of squares calculation unit of the reference sub-area. It accumulates the data of the search sub-region one by one while the reference sub-region data enters the serial shift register group.



Figure 14. Sum of squares calculation unit of the reference sub-area.

3.3. Sum of Squares of All Pixels in a Search Sub-area

The corresponding local matching template is divided as shown in Figure 3. The size of the full template is 31×31 , and R0, R1, R2, R3, and R4 are all rectangular areas of size 7×7 . Different from the reference sub-area, the data in the search sub-area register group is dynamically changed. Therefore, the parallel pipeline structure is used to calculate the sum of squares of the pixels in the search sub-area. The five regions in the local template are divided into three parts for parallel calculation, which are the sum of squares of the regions R1R2, R0, and R3R4.

Figure 15 shows the circuit for calculating the sum of squares of R1R2. The circuit implements the sum of squares calculation of the first 7 lines data in the template and the sum of squares of the region R1R2 in a single clock cycle. First, the 7-way parallel data is calculated by a 8-bit multiplier for the square of each data. Then the summation calculation is done by the pipeline adder. The result is serially stored in 31 20-bit shift registers. Finally, the sum of squares of the region R1R2 (Sum_R1R2) is calculated by the pipeline adders 1, 2 and the adder A, and the sum of squares of all 7 rows (Sum1) in the window can be calculated by the pipeline adder 3 and the adder B.



Figure 15. Calculating the sum of squares of R1R2.

A similar circuit structure is used to obtain the sum of squares of the region R0, R3R4 and the 7 rows occupied by them, and the sum of squares of the remaining 10 rows. The sum of squares of a local template can be obtained by summing Sum_R1R2, Sum_R0, and Sum_R3R4. The sum of squares of the full template can be obtained by summing Sum1, Sum2, Sum3, and Sum4. Hence, they will be ready for full template matching which is possible later.

3.4. Sum of Cross-Correlation Product

For the sum of cross-correlation product (sum of product of the pixel in a reference sub-area and the corresponding one in a search sub-area), a large amount of multiplier resources are required to ensure that the calculation is to be completed in one clock cycle. Because of the strategy of the local template matching, the resource consumption of Digital Signal Processing(DSP) slices is reduced significantly at the cost of a little speed loss.

As shown in Figure 16, the hardware structure constructs a multiplier array with the same number of multipliers as the local template pixels, and the reference sub-area and search sub-region register groups are treated as inputs of the multiplier array. In this way, the multiplier array can ensure that the cross-correlation product calculation of all pixel points in the local template is completed in one clock cycle. Finally, the sum of cross-correlation product of the local template can be obtained by summing all product by the pipeline adder. For the pipeline adder, cycles consumed to calculate the first data depend on the series of the pipeline. Then, each calculation would be completed in one clock cycle.



Figure 16. Calculating the sum of cross-correlation product.

3.5. Sum of Product of the Full Template

For the calculation of the correlation coefficient of the full template, we have obtained the sum of squares of the reference sub-areas and search sub-areas. What needs to be calculated is the sum of the cross-correlation products of the full template. However, the calculation of all cross-correlation product cannot be completed in one clock cycle with existing multiplier arrays. Therefore, this paper uses the multi-clock cycle strategy to calculate the cross-correlation product of the full template. As shown in Figure 17, the remaining pixels of the full template are divided into three parts. When the size of the template changes, the system will still calculate the sum of squares of the reference sub-area and calculate the sum of cross-correlation product correctly. What we need to do is adjust the regional division for time-sharing calculation to ensure that the multiplier arrays can complete full template calculations in three clock cycles.

When the full template correlation coefficient needs to be calculated, the system will make the full template calculation flag to be equal to 1. This signal will cause the cache unit to stop outputting data. The data of the parallel shift register group is held for three cycles, so that the multiplier array is idle to process the product of the full template data. The calculation would be completed in three cycles. Figure 18 shows the state machine that controls the multiplier array. The operation of each state is shown in Table 1.



Figure 17. Regional division for time-sharing calculation.



Figure 18. State machine that controls time-sharing calculations.

Table 1. Resource consumption in field programmable gate array (FPGA).

S0 Calculate the product of the search sub-region and the reference sub-region of the loc	al template.
S1 Calculate the square of the reference sub-region of the local template.	_
S2 Calculate the product of the first part of the full template.	
S3 Calculate the product of the second part of the full template.	
S4 Calculate the product of the third part of the full template.	

3.6. Selection of Adaptive Threshold

Under ideal conditions, the best matching point should be the one with the correlation coefficient closest to 1. There may be some deviations in consideration of the effects of noise and exposure. But the best match point is still one of the points where the threshold is closest to 1. We count all points with

the correlation coefficient from 1 to 0 by the histogram and retain those points close to 1. As shown in Figure 19, the work of Dpram is to count the number of correlation coefficients in different intervals. $1 \sim 100$ address values of Dpram represent the correlation coefficient value of $0.01 \sim 1$. The calculated correlation coefficient multiplied by 100 is used as the read address of Dpram. The value in this address of Dpram is incremented by 1. After the local template completes the calculation of the entire search area, the values in Dpram are taken one by one to be summed from the largest address to the minimum address. Accumulation stops when the accumulated value reaches 8% of the total amount, and the correlation coefficient corresponding to the current address is used as the threshold. This percentage value is configurable. It depends on the specific application scenario. Its determinants are mainly noise and exposure. If the noise or the exposure is high, it can be increased to include more points. Inevitably, increasing this value will result in an increase in the amount of calculation. In most cases, such as the conditions of the experiments in this paper, 8% is sufficient.





4. Simulation and Verification of Hardware Design

4.1. Simulation Results and Analysis

The size of the reference sub-area (calculation window) is an important factor affecting the accuracy of the entire pixel search. If the calculation window is too small, the amount of information that contains image features is small. Although the calculation speed will be increased, the correlation coefficient between adjacent points will have obvious mutations. Correspondingly, if the size of the sub-region is too large, the amount of information that contains image features is large, which can compensate for the effects of various introduced noises, but the calculation time also increases. Therefore, choosing a suitable size of the calculation window is a prerequisite to ensure the correctness of other series of experiments. In this section, the influence of the calculation window on the correlation coefficient is analyzed through experiments. The experiment collected two speckle patterns before and after the deformation by charge coupled device(CCD) camera. The speckle pattern pixel size is 1080 × 800. The horizontal displacement and vertical displacement of the full-pixel have been determined in advance. A search window of a different size is selected in the reference image for search in the target image. The correlation coefficients are calculated to obtain a correlation coefficient grid maps for several different calculation windows as shown in Figure 20.



Figure 20. Correlation coefficient grid diagram of different sizes.

From the above grids, we can conclude that when the size of the calculation window is less than 31×31 , the fluctuation of the correlation coefficient is relatively large. The correlation coefficient has a small difference from other peaks. The main peak is not obvious, and it is easily disturbed by noise and other factors. When the size of the calculation window is above 31×31 , the fluctuation of the correlation coefficient is relatively flat, and as the calculation window increases, the variation range of the correlation coefficient also tends to be gentle and the accuracy of the calculation also increases. As the calculation window continues to increase, the correlation calculation amount also increases sharply, however, the calculation accuracy does not increase significantly as the calculation window increases. Therefore, it is appropriate to select a search window size of 31×31 or 41×41 . The time for search is relatively short, while ensuring accuracy. Two sets of experiments are performed on the proposed fast algorithm. In the first set of experiments, the template size is 31×31 and the search area size is 256×256 . In the second set of experiments, the template size is 61×61 and the search area size is 256×256 . Each group of experiments conducts a search for different numbers of matching points. The time consumed for search with different number of matching points is counted and compared with other algorithms. The results are shown in Table 2.

Algorithm	Search Area Size	Template Size	Search Time (s)			
Algorithm			1 Point	64 Points	100 Points	
NCC [19]	256×256	31×31	4.04	221.56	322.27	
		61 imes 61	15.96	1087.53	1523.28	
NCC-BF [20]	256×256	31×31	2.63	159.26	228.56	
		61 imes 61	12.65	801.34	1286.25	
This paper	256×256	31×31	1.36	14.87	22.76	
		61 imes 61	5.01	69.15	101.41	

Table 2. Comparison of time consumption for three matching algorithms.

As shown in Table 2, we can conclude that the speed of the algorithm proposed in this paper is slightly better when performing single-point matching, which is about three times that of the NCC algorithm [19] and twice that of the NCC-BF algorithm [20]. When the template size is 31×31 , its matching speed is about 15 times that of the NCC algorithm and 11 times that of the NCC-BF algorithm. When the template size is 61×61 , the matching speed is about 16 times that of the NCC algorithm and 12 times that of the NCC-BF algorithm. Therefore, the methods of adaptive threshold and adaptively adjusting the search areas adopted by this algorithm are suitable for multi-point matching search. The larger the template is, the greater the advantage is.

4.2. FPGA Verification Results

This experiment uses the Stratix IV series FPGA of company Altera. Using Quartus II to synthesize the sub-pixel search module, the circuit resource obtained through the auto place & route tool is shown in Table 3.

Resource	Consumption
Combinational Adaptive Look-Up Tables(ALUTs)	8.55%
Dedicated logic registers	5.96%
Total pins	6.08%
Global Clock Buffer(BUFG)	18.75%
Total block memory bits	0.35%
DSP block 18-bit memory bits	28.42%

Table 3. Resource consumption in FPGA.

As shown in the Table 3, the hardware consumption is relatively small while ensuring higher speed. In FPGA verification, the coordinates of the matching points of the full-pixel search and sub-pixel search [21] are input to the serial port debugger through the serial cables and saved as text information. Comparing this result with the results of MATLAB, the two are completely consistent, indicating that the circuit design of the digital speckle correlation method is correct on the FPGA.

The operating time obtained by the logic analyzer is shown in Table 4. The results show that the speed of hardware design proposed in this paper is about 2000 times faster than software when performing single-point matching. The speed of hardware is 600 times faster than the software when performing multi-point matching. It only takes 520 microseconds when the hardware performs a single-point match. It's only a few tens of milliseconds when doing a 100-point match. Therefore, the hardware design of this article achieves the effect of real-time processing.

Platform	Experiment Platform	Search Area	Template Size	Matching Time		
				1 Point	64 Points	100 Points
Software	Intel-i5 2400 3.1GHz	256×256	31 imes 31	1.36 s	14.87 s	22.76 s
FPGA	Altera Stratix IV	256 imes 256	31×31	0.52 ms	23.09 ms	37.62 ms

Table 4. Comparison of simulation time between hardware and software.

5. Conclusions

In this paper, a fast full-pixel search method suitable for digital speckle correlation method is proposed based on NCC correlation matching algorithm. The method first uses the local matching template to eliminate the number of non-matching points to avoid the calculation of unrelated points, thereby improving the calculation speed and efficiency of the algorithm. Then, the histogram statistics are used to select the threshold adaptively. The effects of image noise and local exposure on the algorithm are reduced, which improves the flexibility and robustness of the algorithm. Later, the adaptive adjustment of the search area is used to adjust the position and size of the search area. The calculation speed and computational efficiency are greatly improved without loss of accuracy. Finally, a hardware circuit implementation is realized for the fast integer pixel search algorithm. The circuit can be applied to a real-time image processing system with small hardware cost while ensuring a small loss of search speed.

Author Contributions: Conceptualization, G.Z.; Data curation, Z.K. and K.H.; Formal analysis, G.Z.; Funding acquisition, F.L.; Investigation, S.W.; Methodology, S.W.; Project administration, C.-F.Y.; Resources, C.-F.Y.; Validation, Z.K.; Writing—original draft, K.H.; Writing—review and editing, Z.K.

Funding: This research was funded by the National Natural Science Foundation of China under Grant 61474093.

Conflicts of Interest: The authors declare no conflict of interest.

References

- 1. Dong, H.; Zhou, Y.; Guo, J.; Zhang, W.Y.; Wang, W.S. Digital Speckle Pattern Interferometry for Deformation Measurement. *Acta Photonica Sin.* **2010**, *39*, 19–22. [CrossRef]
- 2. Chen, Z.X.; Liang, J.; Guo, C. Application of digital speckle correlation method to deformation measurement. *Opt. Precis. Eng.* **2011**, *19*, 1480–1485. [CrossRef]
- 3. Yu, G.; Wang, S.G.; Yu, J.H. Technology of digital speckle pattern interferometry and its applications. *Laser Technol.* **2002**, *26*, 237–240. [CrossRef]
- 4. Xiao, Q.Y.; Li, J.; Wu, S.J.; Yang, L.X.; Dong, M.L.; Zeng, Z.M. Denoising of DSPI phase map using sine-cosine filtering and signal energy. *Opt. Precis. Eng.* **2018**, *4*, 936–943. [CrossRef]
- 5. Sun, T.; Liang, J.; Cai, Y.; Wang, Y.Q. Measurement of deformations for copper/aluminum clad under tension with digital speckle correlation. *Opt. Precis. Eng.* **2012**, *20*, 2599–2606. [CrossRef]
- 6. Krehbiel, J.D.; Lambros, J.; Viator, J.A.; Sottos, N.R. Digital Image Correlation for Improved Detection of Basal Cell Carcinoma. *Exp. Mech.* **2010**, *50*, 813–824. [CrossRef]
- 7. Lyons, J.S.; Liu, J.; Sutton, M.A. High-temperature deformation measurements using digital-image correlation. *Exp. Mech.* **1996**, *36*, 64–70. [CrossRef]
- 8. Anwander, M.; Zagar, B.G.; Weiss, B.; Weiss, H. Noncontacting strain measurements at high temperatures by the digital laser speckle technique. *Exp. Mech.* **2000**, *40*, 98–105. [CrossRef]
- 9. Hu, Y.; Wang, Y.H.; Bao, S.Y.; Hu, H.R.; Yan, P.Z. Optimal imaging of digital image correlation speckle under high temperature. *Chin. Opt.* **2018**, *11*, 728–735. [CrossRef]
- Wu, S.J.; Yang, J.; Pan, S.Y.; Li, W.X.; Yang, L.X. Dynamic Deformation Measurement of Discontinuous Surfaces Using Digital Speckle Pattern Interferometry and Spatiotemporal Three-dimensional Phase Unwrapping. *Meas. Sci. Technol.* 2018, *2*, 121–129. [CrossRef]
- 11. Pan, B.; Qian, K.; Xie, H.; Asundi, A. Two-dimensional digital image correlation for in-plane displacement and strain measurement: A review. *Meas. Sci. Technol.* **2009**, *20*, 152–154. [CrossRef]

- 12. Wang, Y.H.; Liang, H.; Wang, S. Advance in digital speckle correlation method and its applications. *Chin. J. Opt.* **2013**, *6*, 470–480. [CrossRef]
- 13. Sun, M.L.; Zong, G.H.; Bi, S.S.; Yu, J.J. Characteristics of Independence on Image Gray Level in Pattern Matching Algorithm and its Application. *Acta Photonica Sin.* **2009**, *38*, 435–440. [CrossRef]
- 14. Singh, S.; Shekhar, C.; Vohra, A. FPGA-Based Real-Time Motion Detection for Automated Video Surveillance Systems. *Electronics* **2016**, *5*, 10. [CrossRef]
- 15. Bravo-Muñoz, I.; Lázaro-Galilea, J.L.; Gardel-Vicente, A. FPGA and SoC Devices Applied to New Trends in Image/Video and Signal Processing Fields. *Electronics* **2017**, *6*, 25. [CrossRef]
- 16. Schreier, H.W.; Braasch, J.R.; Sutton, M.A. Systematic errors in digital image correlation caused by intensity interpolation. *Opt. Eng.* **2000**, *39*, 2915–2921. [CrossRef]
- 17. Huang, H.T.; Fiedler, H.E.; Wang, J.J. Limitation and improvement of PIV. I—Limitation of conventional techniques due to deformation of particle image patterns. *Exp. Fluids* **1993**, *15*, 168–174. [CrossRef]
- 18. Zhang, X.G.; Wang, M.J.; Wang, Y.J. Three-layer fast matching algorithm based on adaptive selection of threshold. *Opto-Electron. Eng.* **2005**, *32*, 83–85. [CrossRef]
- 19. Zitová, B.; Flusser, J. Image registration methods: A survey. Image Vis. Comput. 2003, 21, 977–1000. [CrossRef]
- 20. Sun, C. Fast Algorithm for Local Statistics Calculation for N-Dimensional Images. *Real-Time Imaging* **2001**, *7*, 519–527. [CrossRef]
- Kuang, Z.J.; Zhang, G.H.; Liang F. Fast Hardware Implementation of Sub-pixel Search Algorithm for Digital Speckle Correlation Method. In Proceedings of the 2018 3rd International Conference on Electrical, Automation and Mechanical Engineering, Xi'an, China, 24–25 June 2018; Atlantis Press: Paris, France, 2018; pp. 356–360, ISBN 978-94-6252-538-2.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).