

Development of a compact low coherence interferometer based on GPGPU for fast microscopic surface measurement on turbine blades

Yinan Li^{*a}, Markus Kästner^a and Eduard Reithmeier^a

^aGottfried Wilhelm Leibniz Universität Hannover, Institute of Measurement and Automatic Control, Nienburger Str. 17, 30167 Hannover, Germany

ABSTRACT

Vertical scanning interferometry (VSI) techniques are widely used to profile microscopic surface structures of industrial products. This paper introduces a high-precision fast optical measurement system with an optimized small sensor head for the measurement of precision surfaces on a turbine blade or blisks (blade integrated discs). The non-contact measurement system is based on a low coherence interferometer (LCI), which is capable of fast profiling of 3D sample surface with a nanometer resolution and has a larger measurement range compared to conventional microscopes. This results in a large amount of sampled data and a high computational time for the evaluation of the data. For this reason, the used evaluation algorithm in this paper is accelerated by the Compute Unified Device Architecture (CUDA) technology, which allows parallel evaluation of the data stack on independent cores of a General Purpose Graphics Processing Unit (GPGPU). As a result, the GPU-based optimized algorithm is compared with the original CPU-based single-threaded algorithm to show the approximate 60x speedup of computing the Hilbert Transformation, which is used to find the depth position in the correlogram of each pixel of the sampled data. The main advantage of the GPU computing for the evaluation algorithm of the LCI is that it can reduce the time-consuming data evaluation process and further accelerates the whole measurement.

Keywords: Non-contact measurement system, Interferometer, Low coherence interferometry, CUDA

1. INTRODUCTION

The microscopic surface inspection of engine components is a relevant prerequisite for aviation maintenance [1] and the optimization of the aerodynamic performance [2]. In this paper we present an in-house developed optical surface measurement system based on a low coherence Michelson interferometer, which allows a large scanning range using a telecentric lens with a large field of view (FOV) and a wide depth of field (DOF). Therefore, the low coherence interferometer (LCI) system is well suited for the measurements of microscopic structure on curved surfaces, like turbine blades.

For the application, the LCI measurement system has the advantages of non-contacting and high-precision data acquisition but the challenge of a huge computing time of the evaluation algorithms for a large measurement volume. As a simple method to reduce the computing time the data reconstruction can be parallelized and many-core CPU can accelerate the evaluation algorithms. In recent years, the programmable GPU-based general computation technology shows a higher performance with massive threading and high memory bandwidth for parallel processing, which allows independent analysis and evaluation of the sample data. Compared to the common supercomputing, a GPU-based desktop PC is a low cost alternative for a massively parallel system. For science and engineering applications the Compute Unified Device Architecture (CUDA) technology developed by NVIDIA published in [3] is widely used as a cost-effective and easy-programming parallel computing platform. Many applications based on CUDA were explored in image processing for off-axis quantitative phase [4], 3D surface measurement and visualization with real-time quadrature transform method [5] and 3D numerical simulation for the attosecond physics regime [6].

Moreover, an approximate application of the data processing to LCI using CUDA has also been presented in [7]. An evaluation algorithm based on the least-squares polynomial curve fitting was used in this article for accelerating data evaluation process. In order to further optimize the computational efficiency we implement a Butterworth bandpass filter for noise elimination and an evaluation algorithm based on the Hilbert Transform (HT) method [8] utilizing the GPU computing. The main measurement software is developed in the Python programming language and connected via Pycuda (Nvidia's CUDA parallel computation API from Python) to the GPU-based algorithm in C language.

*yinan.li@imr.uni-hannover.de; phone +49 511 7623236

2. EXPERIMENTAL SETUP

The experimental setup of the measurement system is based on a conventional Michelson configuration as shown in Figure 1, where the interference pattern can be caused by the different path lengths of the orthogonal reference and measurement paths. The reference mirror is driven by a precision piezo stage (Newport NPX400SG) with a minimal step length of 8 nm by closed loop control. The system uses a deep red Power-light emitting diode (LED) light source with a wavelength of 645 nm, which outputs high brightness and a low coherence length in nature. The intensity of the reflected light from sample surfaces depends upon the material and the surface roughness. In most situations the intensity is lower than the reflected light from the reference path. For this reason, a variable neutral density filter is positioned in front of the reference mirror to optimize the contrast of the interference pattern for the image recording.

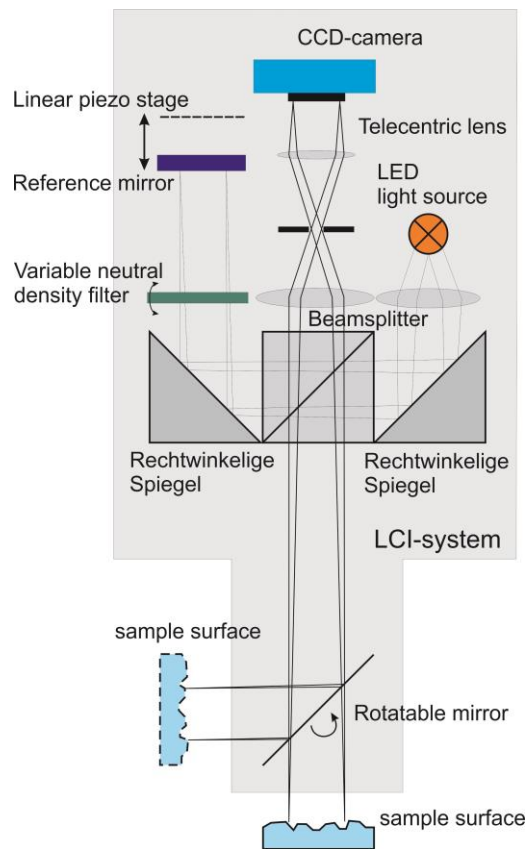


Figure 1. Schematic of LCI based on a conventional Michelson configuration.

In order to reduce the overall size of the system, the reference path and the light source path are rotated against the optical axis of the beam splitter by reflectors. A charge coupled device (CCD) camera with a 4x telecentric lens which is capable of recording the images at a relatively large focus distance of 110 mm, is used in the system. With the help of a rotatable mirror at the sensor head, the system can further measure an object surface flexibly parallel or orthogonal to the optical axis. Therefore it is suitable to measure surfaces of complex shaped work pieces, for example the blade's surface between two blades in the blisks (blade integrated disks).

The Figure 2 shows a schematic diagram of the signal flows and data flows in this LCI-system. In the measurement procedure, an Arduino microcontroller synchronizes the camera shutter and the movements of the piezo stage. A piezo controller drives the stage with a nanoscale scan step width (Δd) in a closed loop mode for high motion accuracy. Once the length of the reference path is changed in one step, the image acquisition is triggered at this position ($d = \Delta d \cdot n$) and the data is stored in a data stack (3D Matrix M) shown in Figure 5Figure 4. Here n is the step number of N steps in

the whole scanning process. The observed data stack will be evaluated pixel by pixel using the HT to find the central fringe position in z direction within the image stack. And then the central fringe can be converted into the relative surface height of the pixel using the known scanning position d .

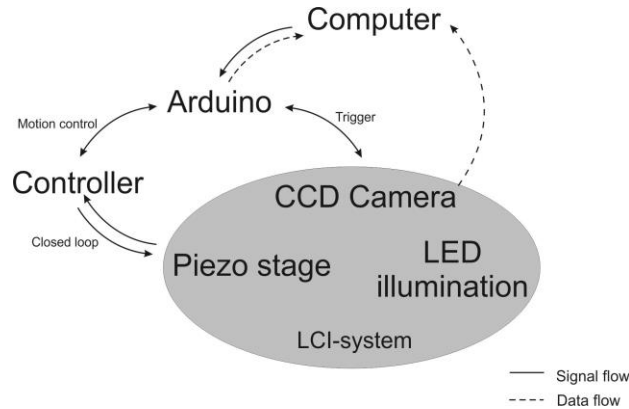


Figure 2. A schematic diagram of the signal flows and data flows of the LCI-system

3. DATA PROCESSING USING CUDA

3.1 Evaluation algorithm

To detect the central fringe for an interferogram, many evaluation algorithms have been developed in the past [9, 10, 11]. The algorithms used different methods like polynomial interpolation, Fourier method and wavelet method. In this paper we use the HT to find the interferogram envelope and then to determine the peak of this envelope.

The interferogram of each pixel is filtered with a bandpass filter firstly to reduce the electronic pixel noise of the CCD camera. The filter is set according to the carrier frequency, which can be calculated using the scan step width and wavelength of the light source. Because the camera shutter and the movements of the piezo stage must be synchronous, a regular time interval between two frames cannot be guaranteed. For this discrete-time sampled interferogram we use a discrete Hilbert transform, whose function $H(u)[n]$ is written as

$$H(u)[n] = DTFT^{-1} \{U(w) \cdot \sigma_H(w)\} \tag{1}$$

where the $U(w)$ is the value of the discrete-time Fourier transform (DTFT) of the input data $u[n]$ from each interferogram, n is the number of the frame and $\sigma_H(w)$ is a signum function in the form:

$$\sigma_H(w) = \begin{cases} e^{+i\pi/2}, & -\pi < w < 0 \\ e^{-i\pi/2}, & 0 < w < \pi \\ 0, & w = -\pi, 0, \pi \end{cases} . \tag{2}$$

By taking inverse DTFT of $U(w) \cdot \sigma_H(w)$ the Hilbert transform $H(u)[n]$ can be found. This way, the analytic representation of the Hilbert transform can be defined as

$$\hat{U}[n] = u[n] + i \cdot H(u)[n] \tag{3}$$

where the real part is the input data and the imaginary part is its Hilbert transform. This way, the amplitude $a[n]$ of the analytic signal is calculated as absolute value of the input data and its phase shift with the following form:

$$a[n] = \sqrt{u[n]^2 + H(u)[n]^2} . \tag{4}$$

Figure 3 shows an interferogram and its envelope curve calculated by the HT algorithm (4). Because the envelope curve is smoother than the original interferogram with noise, the maximum position of the interferogram can be estimated.

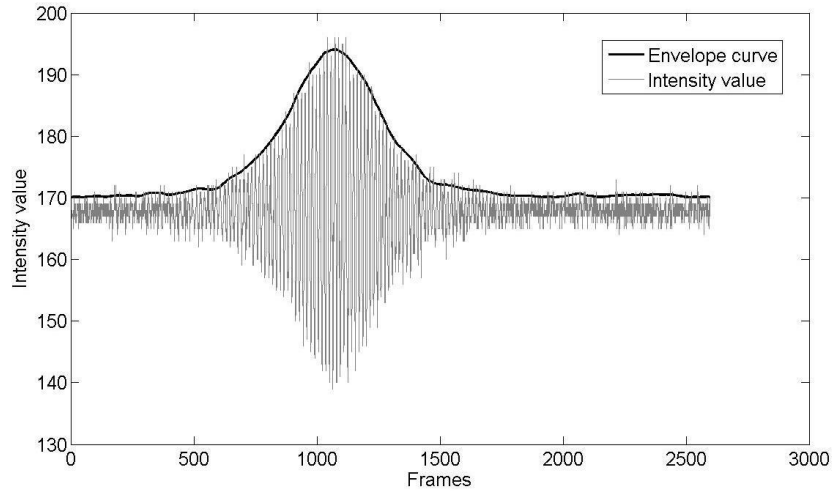


Figure 3. Signal processing of a interferogram using the envelope curve calculated via Hilbert transform.

3.2 Data processing using CUDA

The interference correlogram of the pixels in the data stack are independent of each other in terms of data processing. It is a necessary prerequisite for the parallel computing, because the parallel architecture model of the CUDA technology called Single-Instruction, Multiple-Thread (SIMT) means that multiple threads execute the same instruction on different data. The GPU-based evaluation algorithm using CUDA distributes the independent correlograms into available threads on the GPU shown in Figure 5. The thread that is the least executed unit in GPU has a unique ID and register in the Block. A multithreaded program distributes the data into blocks of threads that execute each other simultaneously. The executed threads in groups start together and individually compose a warp, which is created and scheduled by the GPU-multiprocessor. A thread block that contains the threads has shared memory available to the threads in this block. Using the shared memory the global variables like initial parameters of bandpass filter and HT algorithm can be defined for the threads. The blocks in a grid execute always in the same kernel on the GPU.

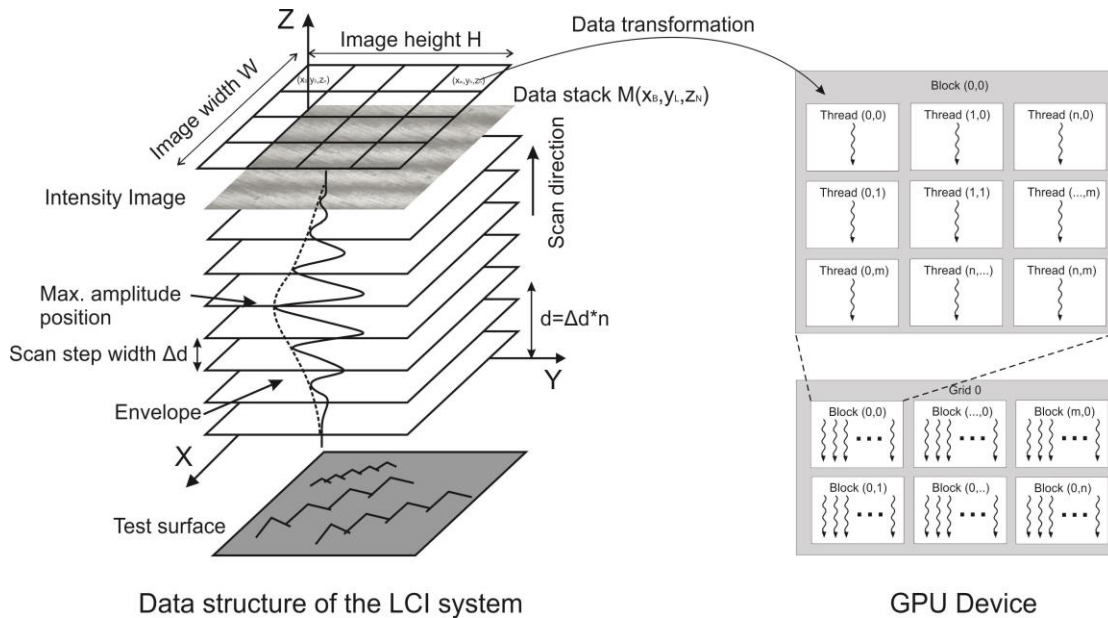


Figure 4. Data structure of the LCI system and the data distribution into the thread block in CUDA programming.

The software structure can be divided into two parts. One standard sequential part is the main program for the measurement system that is developed in Python language; the other parallel part for the GPU-based HT algorithm is programmed in C language. As a bridge the Pycuda combines both parts [12]. In the computational phase, the main program transmits the data stack from the CPU memory (RAM: Random Access Memory) into the global memory of the GPU (VRAM: Video RAM). The data on the VRAM are distributed by the CUDA kernel into GPU-threads and are calculated simultaneously. After the computing the results are transmitted from the VRAM back to the CPU RAM.

The performing experiments using the GPU-based HT algorithm is running on a low-cost lab computer equipped with an Intel Core i5 2400 processor and an NVIDIA GeForce GTX 970 graphics card using the Maxwell architecture. The specifications are shown in Table 1.

Table 1. The basic hardware specifications.

Specifications of the PC system	
PC Processor Frequency	3.1 – 3.4 GHz
PC-RAM	8GB, DDR3
GPU Frequency	1051 MHz
VRAM	4GB, DDR5
GPU Bus Width	256 bit
GPU Drivers Vision	347.88
Compute Capability of CUDA	5.2
CUDA Edition	6.5

4. MEASUREMENT RESULTS AND PERFORMANCE COMPARISON

The LCI system described in section 2 with the telecentric lens with a large FOV and the wide DOF has a relative large measurement volume. Therefore, it is suited for the inspection of curved surfaces. In the following measurement, a microscopic surface of the suction side of the turbine blade without protective layer is scanned and shown in Figure 5. The partial shape of the surface and the machining scratches on the surface caused by a CNC machine are clearly visible. The surface roughness and waviness can be divided from the measurement result using an areal filter after ISO 4287 for further analysis [13].

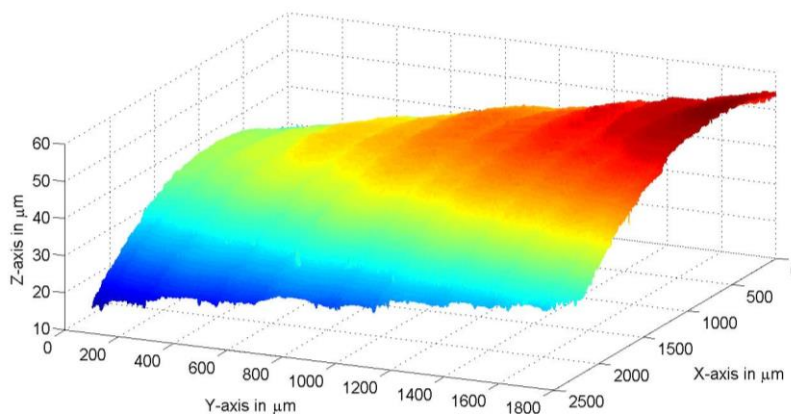


Figure 5 A microscopic surface on the suction side of the turbine blade

The measured surface shown in Figure 5 is evaluated from 1000 frames with resolution of 1388 x 1038 pixels each and a scanning step height of 60 nm. In this case, the used lab computer takes 1932 s in a single CPU-thread or 29.4 s with the GPU-based algorithm in the GPGPU for the data processing. The GPU computing is compared with the single-threaded CPU implementation to show a 65x speedup for this 1.4 GB data stack. The table 2 shows the computational time with CPU and GPU for the different frame sizes. The more frames means a larger vertical measurement range for the LCI-system. The speedup between CPU and GPU goes from 61x for the data stack with 500 frames to 72x for the data stack

with 1500 frames. The advantage of the GPU computing with massive threads increase with the increase in the size of frames.

Table 2. Computing time for single-threaded CPU and GPU implementation.

Number of frames	Single CPU-thread in s	GPU in s
500	1053	17.3
1000	1932	29.4
1500	3167	43.7

5. CONCLUSION AND OUTLOOK

In this article, a GPU-based fast microscopic surface measurement system combined with a HTAlgorithm to allow parallel evaluation of the data stack was presented. The GPU-based measurement data evaluation showed an average 65x speedup for the data stack with 1000 frames compared to a single CPU-threaded implementation.

In order to further reduce the whole measurement time, the LCI system will use a high-speed camera with frame rates up to several hundred instead of the native camera with 30 fps, because the scanning time occupies 30% of the measurement time. Moreover, the LCI system can work together with another measurement instrument like a fringe projection system to detect objects surface in multiscale [14]. In the future, this LCI measurement system will be included in an industrial robot based guided system for the automated measurement of turbine blades.

6. ACKNOWLEDGEMENTS

The authors would like to thank the German Research Foundation (DFG) for funding this project “Multiscale Measurement of Blade Geometries with Robot-Supported, Laser-Positioned Multi-Sensor-Techniques” within the Collaborate Research Center (SFB) 871 “regeneration of complex capital goods” (<http://www.sfb871.de>).

REFERENCES

- [1] Domaschke, T., Schueppstuhl, T., and Otto, M., “Robot Guided White Light Interferometry for Crack Inspection on Airplane Engine Components,” In *ISR/Robotik 2014, 41st International Symposium on Robotics, Proceedings of* (pp. 1-7) VDE (2014).
- [2] Hummel, F., Lätzerich, M., Cardamone, P., and Fottner, L., “Surface roughness effects on turbine blade aerodynamics,” *Journal of Turbomachinery*, 127(3), 453-461 (2005).
- [3] nVIDIA Corporation, “CUDA C Programming Guide” version 6.5, (2015)
- [4] Pham, H., Ding, H., Sobh, N., Do, M., Patel, S., and Popescu, G., “Off-axis quantitative phase imaging processing using CUDA: toward real-time applications,” *Biomedical optics express*, 2(7), 1781-1793 (2011).
- [5] Espinosa-Romero, A., and Legarda-Saenz, R., “GPU based real-time quadrature transform method for 3-D surface measurement and visualization,” *Optics express*, 19(13), 12125-12130 (2011).
- [6] Dziubak, T., and Matulewski, J., “Three-dimensional numerical simulations in attosecond physics regime using the CUDA technology: the stabilization phenomenon,” In *SPIE Optics+ Optoelectronics, International Society for Optics and Photonics*, (pp. 807112-807112). (2011).
- [7] Tomczewski, S., Pakula, A., Van Erps, J., Thienpont, H., and Salbut, L., “Low-coherence interferometry with polynomial interpolation on Compute Unified Device Architecture-enabled graphics processing units,” *Optical Engineering*, 52(9), 094105-094105 (2013).
- [8] Zweig, D. A. and Hufnagel, R. E., “Hilbert transform algorithm for fringe-pattern analysis. San Dieg-DL Tentative,” *International Society for Optics and Photonics*, (1990).
- [9] Tomczewski, S., Pakula, A., Van Erps, J., Thienpont, H., and Salbut, L., “Low-coherence interferometry with polynomial interpolation on Compute Unified Device Architecture-enabled graphics processing units,” *Optical Engineering*, 52(9), 094105-094105 (2013).

- [10] Larkin, K. G., "Efficient nonlinear algorithm for envelope detection in white light interferometry," *JOSA A*, 13(4), 832-843 (1996).
- [11] Recknagel, R. J., & Notni, G., "Analysis of white light interferograms using wavelet methods," *Optics Communications*, 148(1), 122-128 (1998).
- [12] Klöckner, A., Pinto, N., Lee, Y., Catanzaro, B., Ivanov, P., Fasih, A., & Tetali, P. "PyCUDA: GPU run-time code generation for high-performance computing," Arxiv preprint arXiv, 911. (2009).
- [13] ISO 4287, "Geometrical product specification (GPS)—Surface texture: profile method terms, definitions and surface texture parameters," International Organization of Standardization (2000).
- [14] Schlobohm, J., Li, Y., Pösch, A., Langmann, B., Kästner, M., & Reithmeier, E., "Multiscale Optical Inspection Systems for the Regeneration of Complex Capital Goods," *Procedia CIRP*, 22, 243-248 (2014).