Conclusions

1. The multi-channel vision-based and GPU-accelerated concept of improvement of aircraft flight navigation system has been proposed and realized;

2. The methodology of multi-channel visual-based implementation into flying vehicles for getting the pitch and bank angles as well as their reliability evaluation has been developed.

3. The elaborated multichannel & multithreading horizon-detection algorithm has been verified.

References

1. NVIDIA Jetson TX1 Module. Technical specifications (https://developer.nvidia.com/embedded/jetson-tx1)

2. Horizon detection for UAVs with Python (https://github.com/nsbradford/HorizonCV)

3. Cyganski N., Gillette B., et al. The Autonomous Cargo Aircraft Project ACAP, 2017 (https://web.wpi.edu/Pubs/E-project/Available/E-project0427171-.43558/.../Report.pdf)

UDC 531.383

Shkvar Ye.O., E Shi-ju, Kryzhanovskyi A.S., Fang Yuan

GPU-ACCELERATED AND MPI-DISTRIBUTED MULTI-CHANNEL VISUAL EMBEDDED PORTABLE SYSTEM, BASED ON CLUSTERING NVIDIA JETSON NANO COMPUTATIONAL PLATFORM

Introduction. In the past years the development of modern controlling strategies and their efficient engineering implementations is very intensive. This favorable trend stimulates the design and use of various control systems such as Fly-by-wire (FBW) which replace the conventional manual aircraft flight controls with an electronic interface and can handle a large number of emergencies and realize guided driving and even full autopiloting as well as improve the safe functioning of an autopilot of aircraft. Because the application area of such systems in the commercial and military field is pretty high, now they have been widely applied in the different high-speed vehicles such as aircrafts, bullet trains, cars and unmanned aerial vehicles (UAV). Although the aircraft crashes are much less than for other vehicles, especially in a short period of time, there were two incredible aircraft crashes of famous airlines a few years ago. (Lion Air Flight 610 in October,2018 and Ethiopian Airlines Flight ET302 in March,2019), which were the first major accidents with a high death toll involving the new series of Boeing 737 MAX aircraft introduced in 2017. The main reason of both crashes is identical – improper Maneuvering Characteristics Augmentation System (MCAS), so the reliability of autonomous controlling systems is still awaiting improvement.

This article describes the results of research, which further develops the conceptual idea and initial proposal, formulated by authors in the previous study [1] – to offer adding multiple visual channel system as one more very (if not the most) informative source of information, based on portable GPU-accelerated platform for real-time control of movement parameters that can improve the reliability of the movement vehicles control system and make it much more veracious. The engineering basis of previous study [1] was based on embedded NVIDIA Jetson TX1 platform, equipped with several (up to 4) cameras, and the corresponding programming model of simultaneous access to the cameras and their parallel processing was based on the multithreading model of parallelism with CUDA model of GPU acceleration, implemented in Python version of OpenCV library. The current research suggests using a more flexible, scalable, and performant hardware and software framework described below.

The Problem Statement. The concept of improvement of flight controllability and safety is based on: 1) realization of multi-channel continuous automatic visual monitoring of aircraft or spacecraft surrounding; 2) massively-parallel processing of video content, based on Compute Unified Device Architecture (CUDA), GPU-accelerated NVIDIA Jetson platform; 3) independent simultaneous real-time processing of multi-channel video content due to the use of cluster architecture of computational system (please see the Attachment, Fig. 5) and Message Passing Interface (MPI) and 4) positioning of the proposing system as a perspective informative (especially at the most critical flight stages as climb and descent) additional source of flight state information and an additional element of automatic control loop for coordinating the current flight data from other sensitive equipment for a reasonable synthesis of the required control laws.

The most important **goals** of the current research step are to: 1) improve the scalability of the developing approach by replacing the single GPU-accelerated computational platform NVIDIA Jetson TX1 (256 CUDA cores) with a distributed system, having clustering architecture and based on several cheapest platform NVIDIA Jetson Nano (128 CUDA cores); 2) improve the productivity of processing the horizon recognition problem due to its implementation in compiled C programming language instead of interpreted Python; 3) elaborate the efficient MPI-based program code for distributed processing the visual content from different independent sources – cameras; 4) compare the proposed approach of distributed GPU-accelerated visual data processing with its previous multi-threading version and evaluate the relative efficiency.

Architectural and technical implementation of the proposed hardware and software improvements. As it was mentioned above, the principally new concept of improvement of flight controllability and safety, based on realization of multi-channel continuous automatic real-time visual monitoring of high-speed vehicle surrounding, combination of massively-parallel CUDA GPU-accelerated and Message-Passing Interface (MPI) computations, is proposed as a perspective additional source of flight state information and for coordinating the current flight data from other sensors and reasonable synthesizing the required control laws. Previous research results, associated with multithreaded implementation of GPU-accelerated processing of visual content and the developed OpenMP-based algorithm, demonstrated low efficiency losses (up to 6.7%) for GPU-accelerated visual data processing of several (up to 4) independent threads, coming from 4 differently located identical video-cameras using the embedded computational platform NVIDIA Jetson TX1 [1]. Nevertheless, further increasing the number of sources of video-content is limited by the productivity of the single portable computational platform, that should be shared between all attached cameras. That is why in the current stage of research direction we have changed the computing paradigm from OpenMP multithreading to multiprocessing based on a combination of MPI technology for scaling computational resources. As a result, the computational platform was modified to more scalable clustering system, consisting of several (4 pieces at the moment, but there is no limit to their increase) very compact NVIDIA Jetson Nano nodes with 128 GPU cores and 4GB RAM each, linked by Gigabit Ethernet switch (please see the

Attachment, Fig. 5). The triple-channel mode of MPI-based visual-data processing is demonstrated by Fig. (please see the Attachment, Fig. 4). Obviously, the reliability of the proposed clustering system is much higher compared to any multicore computing device, because even if one node of this cluster fails, the rest of nodes will work and interact each other independently, so in this case we simply get a failure of one of several information sources without losing the functionality of the computing device as a whole. In addition, this platform, being built on the NVIDIA Jetson platform, is capable of implementing artificial intelligence logic for real-time visual control systems proposed for development.

As for improving the software, for optimal and most efficient computations, all the program code was implemented in C as the native programming language for the applied computing technologies such as CUDA and MPI.

Testing and comparative analysis of the proposed distributed computational platform efficiency. In order to estimate the efficiency of the proposed computational strategy, we used the previous Python-based multithreading approach [1] as the basis for comparison and obtained new results of testing the computational productivity of the developed C-based MPI-parallelized method. In the last case, all cameras were attached to and independently processed by the corresponding node of the clustering distributed system.

The results of computations show (Fig. 1) that the highly optimized binary C language-based code allows to increase the productivity of serial case (one process) in comparison with one thread execution of Python code, having similar functionality (both have been tested on embedded NVIDIA Jetson Nano platform), in 228/9.6=23.75 times.

As it follows from Fig.1 and Fig.2, a parallel MPI implementation of GPU-accelerated visual data processing, tested on the cluster of 4 embedded NVIDIA Jetson Nano platforms, is significantly faster than Python multithreading code, which has the similar functionality and was tested on the single embedded NVIDIA Jetson Nano platform (in 84.3 times for the case of 4 visual threads processing).

In addition, a parallel MPI realization of clustering computations is much more reliable and fault-tolerant in comparison with multithreading computing model. Failure of any node of distributed clustering system will not affect the functionality of the rest of nodes, but it requires some special mode of clustering system operating. In the latter case, it is necessary to run the task not as one with several MPI processes, but as a package of several single-process tasks distributed over the nodes of the cluster system.

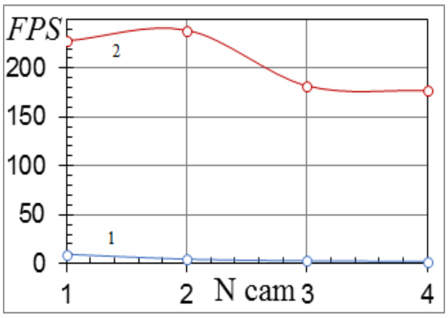

Fig. 1. Dependence of productivity reduction of GPU-accelerated embedded NVIDIA Jetson Nano platform in Frames Per Second (FPS) for multithreading in Python (line 2) and MPI technology in C realization (line 1) in case of the same testing video-content

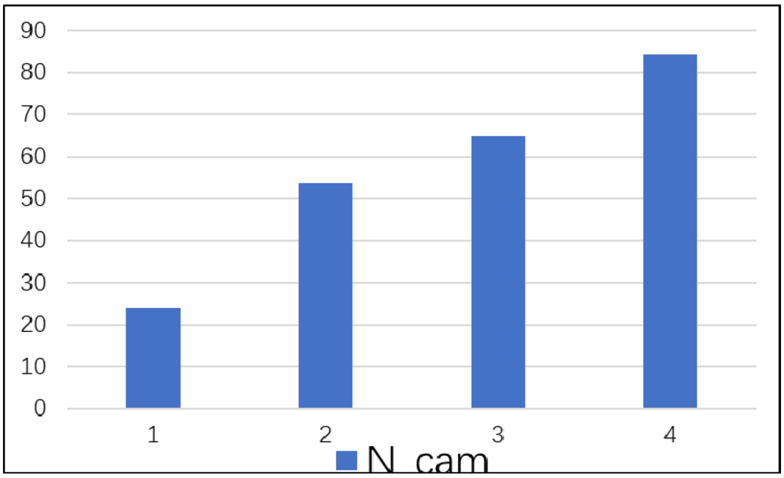

Fig. 2. Dependence of clustering MPI-based productivity related to multithreading visual data processing vs. number of independent visual–threads (cameras)

Fig.3 compares the performance loss, associated with different models of parallelism realization (data exchange between different processes vs. simultaneous processing of several threads). As it can be seen, MPI realization on clustering system with GPU-accelerated nodes for the case of 4 visual threads processing allows to achieve 22.3% of performance loss for each of corresponding computational processes in comparison with multithreading technology of visual data processing by the single NVIDIA Jetson Nano platforms, where the performance loss is 78%, so the clustering technology utilizes the available productivity more efficiently due to much better performance scalability.

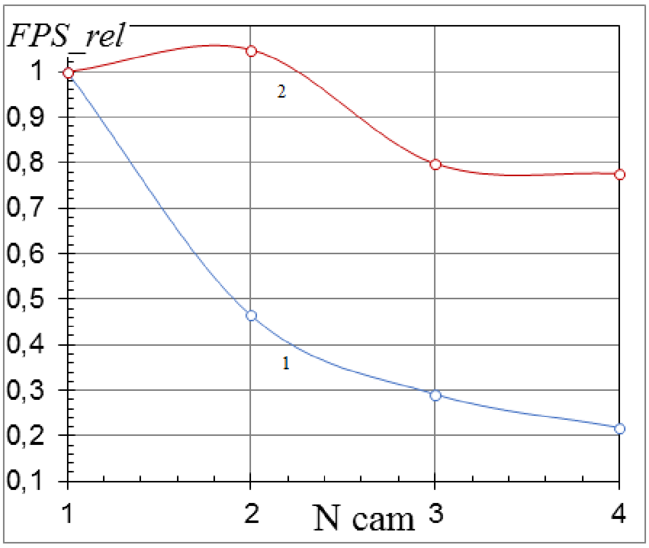

Fig. 3. Dependence of productivity reduction of GPU-accelerated embedded NVIDIA Jetson Nano platform in relative FPS (FPS/FPS_1) for multithreading in Python realization (line 1) and MPI technology in C realization (line 2) in case of the same testing video-content

The typical electrical power consumption of the NVIDIA Jetson Nano platforms is 5W and in maximum mode up to 10W. In case of multithreading parallelism, the maximum mode was used (single system has processed up to 4 independent video-threads), so 10W, but in cluster realization each of its nodes processes only one video-threads, so typical mode 5W was enough. Thus, the 4-node clustering system utilizes 5x4=20W electrical power, twice of maximum mode of a single system (20W/10W=2), but computational efficiency of distributed MPI computations increases up to 84.3 times. In addition, independent processing of video-content by different threads provides substantially better scalable reliability and fault-tolerance.

Conclusions

1. A multichannel concept for improvement of the aircraft flight navigation system with GPU acceleration and MPI scaling is proposed and implemented.

2. The realized MPI scaling of NVIDIA Jetson GPU-accelerated real-time massively-parallel processing of video-content is much more efficient and reliable in comparison with previously realized multithreaded version, so it can be recommended for use in various modern high-speed vehicles.

References

1. Shkvar Y, Dai Z, Shi-Ju E, et al. A New Concept of GPU-Accelerated On-Board Visual System for Aerial Vehicles. [J] *Engineering Science & Technology*, 1(2), 2020, – pp. 84-93 doi.org/10.37256/est.122020451

2. NVIDIA Jetson Nano platform specifications: developer.nvidia.com/embedded/jetson-tx1

УДК 531.36, 629.7

Quijada Pioquinto J.G.

GENERATION OF A DATABASE OF AIRFOILS WITH BEZIER-PARSEC PARAMETERS

Bezier-PARSEC parameters are used to describe profile geometry in aerodynamic shape optimization processes due to the reduced number of design variables and their ability to simulate airfoils of multiple families. This favorable feature is ideal for standardizing a set of profiles to be used in deep neural network training. The main challenge of creating a data set for training a neural network is to obtain a lot of data with the same parameters to describe it. This paper offers a proposal to generate a database of profiles with Bezier-PARSEC parameters, using the algorithm of Adaptive Differential Evolution based on the History of Success and methods of population size reduction.

Keywords: SHADE algorithm, parameterization Bezier-PARSEC, population size reduction methods.

Introduction. In 2004, Rogalsky in his doctoral thesis proposed the new Bezier-PARSEC (BP) parameterization method for airfoils [1], [2]. This new method aims to accelerate the convergence of evolutionary aerodynamic optimization processes. Rogalsky and Derksen proved the abilities to represent differents airfoils with BP parameters (symmetrical and asymmetrical airfoils, high and low lift airfoils, wing airfoils and turbomachinery use airfoils). These tests were performed with the differential evolution (DE) algorithm, and the objective function was the geometric deviation [3], [4]. The results were satisfactory, showing that one could have a small number of design variables