UDC 531.383

Shkvar Ye.O., E Shi-ju,
Kryzhanovskyi A.S., Cai J.C., Dai Z.

MULTI-CHANNEL GPU-ACCELERATED VISUAL HORIZON RECOGNITION, BASED ON NVIDIA JETSON COMPUTATIONAL PLATFORM

Introduction. The conceptual idea and initial proposal of this study is to offer adding multiple visual channel of information as one more very (if not the most) informative source of control of movement parameters that can improve the stability and reliability of the movement vehicles control system and make it much more veracious. This obvious at a first glance solution could not be implemented earlier, since it requires continuous realization of large volume of very productive on-board computations in real-time mode and with quite small energy consumption. Last hardware and software decisions of NVIDIA are concentrated to implementation of their Compute Unified Device Architecture (CUDA)-based model of massively parallel computations to developing and implementation of the artificial intelligence technologies as well as to developing the portable realizations of these promising architectural achievements in the family of NVIDIA Jetson computational devices, oriented to computer vision technologies. Technologically NVIDIA supports the importance of sensors redundancy – the latest modification of the simplest and smallest realization of NVIDIA Jetson platform – Jetson Nano B01, issued in early 2020, was equipped with two MIPI CSI slots for in-built cameras.

The formulated in this paper engineering conceptual idea has been realized by developing the portable embedded system on the base of one of NVIDIA embedded NVIDIA Jetson platforms – the Jetson TX1, which is portable (87x50 mm) and has low level of power consumption (under 10W) and supercomputing productivity (over 1 TFLOPs computational performance, up to 1400 Mpix/s camera interface capability) [1]. At the same time this platform is associated with the computational CUDA-based programming environment, that is very flexible and suitable for real-time massively-parallel processing (256 cores NVIDIA Maxwell GPU) of any information and, in particular, visual content, similar to the video processing by brain of biological objects which is associated with the artificial intelligence technology. Therefore, this platform is able to implement some elements of artificial intelligence logics for the proposed for developing real-time visual systems of control.

The Problem Statement. The concept of improvement of flight controllability and safety, based on: 1) realization of multi-channel continuous automatic visual monitoring of aircraft or spacecraft surrounding; 2) massively-parallel GPU-accelerated NVIDIA Jetson platform; 3) CUDA and Artificial Intelligence data processing and 4) functioning in real-time mode, is proposing as a perspective additional source of flight state information and for coordinating the current flight data from other sensors for reasonable synthesizing the required control laws.

The goal of this research was to analyze the advantages and evaluate feasibility and expediency of the proposed concept of visual multichannel processing support as an additional element of automatic control loop for aircraft/spacecraft flight safety and reliability improvement at the most critical flight stages as launch and landing.

Software and hardware aspects of single- and multi-channel visual content processing approach for aerial vehicle orientation recognition. Vision-based systems can evaluate an aircraft's orientation directly with respect to the ground and, in particular, the bank angle (Φ) and pitch angle (Θ). There are many different approaches and algorithms with various accuracy and complexity levels, but at the current initial stage of study our aim is to choose one of the simplest workable solution that can take a benefit from its realization on the GPU-accelerated platform (OpenCV library and CUDA), that was clearly demonstrated by Nicholas Bradford in his Python-based project for Unmanned aerial vehicles «Autonomous Cargo Aircraft» [2, 3], so the described algorithm was adapted, expanded by us and then tested on the NVIDIA Jetson TX1 platform.

The basic principle of the chosen algorithm is to describe both the sky and the ground as a statistical distribution in color space. Then the task is to find the best set of points for a particular distribution in the image. Knowing that we can use a line to approximate the horizon, we can limit our search to a set of points that consist of a continuous area separated by a straight line in the image.

This method can be illustrated by fitting two polygons to the sky and ground areas at the same time. For a straight-line horizon, the sum of the edges of the two polygonal regions will always equal 8. On the other hand, more computationally intensive algorithms can use the same method to fit N-sided polygons into two regions and describe a non-linear horizon.

The platform system we use is Ubuntu 18.04 and using JetPack provided by NVIDIA which supports CUDA 9.0 and we used OpenCV library for video-processing to run the Python code. Simply put, having OpenCV installed makes it easier to write code to facilitate the procedure of pre-processing images prior to feeding them into deep neural networks. Furthermore, in a GPU-enabled CUDA environment, there are a number of compile-time optimizations we can make to OpenCV, allowing it to take advantage of the GPU for computation accelerating.

Results of testing the single- and multi-channel horizon-detection algorithms. Several single-channel results of running the Python code in its original version [2] on the NVIDIA Jetson TX1 platform are illustrated by Fig. (please see the Attachment, Fig. 2). The obtained average results of comparison of NVIDIA Jetson TX1 GPU processor (5.6 frames per second (fps) and CPU Intel I5 5257U (3.9 fps) demonstrated the real GPU acceleration of the computations about 1.44 times. It shows that massive parallel computation on multi-core GPU chip even in case of quite weak GPU (only 256 cores) is both effective and reliable for visual data processing.

Technologically, for independent multi-channel horizon-detecting visual processing realization, two USB cameras were connected to NVIDIA Jetson TX1 or for their simulation two video-files were opened simultaneously and the software code was modified on the base of threads, which implement the horizon-detecting algorithm and process the data from different cameras independently and in parallel. This approach looks the most universal, because it doesn’t have any limitations related to the number of video-data channels (cameras), excepting limited resources of RAM and performance of computational platform that will be shared by all generated threads. In the framework of this study, we restrict our analysis to considering only a two-channel case based on two identical cameras with the same orientation. But for testing purposes we tested two– three and even quad-channel cases, where video-content has been simulated by several slightly corrected versions of the same chosen video-file (modifications were made by slightly change of contrast).

An error estimation algorithm and matching measurements of pitch and bank angles. Three-level classification of visual system functioning. In case of more than one source of video content it is possible to evaluate the level of mismatch of stream values of evaluated angles from all video-channels. Let's consider N video-channels and any *i*-th thread (*i*=0,…, N), associated with corresponding source of video-data, generates the values of pitch θ_i and bank γ_i angles for any frame of video-stream. The maximum absolute errors for these angles can be found as $\varepsilon_\theta = \max(\theta_i) - \min(\theta_i)$ and $\varepsilon_\gamma = \max(\gamma_i) - \min(\gamma_i)$ respectively. Based on these errors, three modes of the visual system state were introduced, namely: stable, warning and unstable.

Stable mode. Here we suppose that the results from all video-sources matched in case if the following conditions are both satisfied: $\varepsilon_\theta \leq \varepsilon_{max}$; $\varepsilon_\gamma \leq \varepsilon_{max}$, where maximum allowable discrepancy was taken by us as ε_{max}=2°. If it is true for the current video-frame and in addition for all m previous frames, in this the most favorable case we suppose that the whole multi-channel video system generates the reliable data stream and it can be used by the system of aerial vehicle control. On the screen the developed multi-threading software code illustrates this reliable condition of the visual system by the green frames of the images from all video-data sources (please see the Attachment, Fig. 3).

Unstable mode. In case if at least one of the mentioned above conditions fails, the system is considering as unstable working and the obtained evaluations of pitch and bank angles can't be applied for aerial vehicle control. On the screen this unstable state of the visual system is illustrated by the red frames of the video images (please see the Attachment, Fig. 4).

Warning mode. For providing additional data confidence the transient warning state from unstable to stable modes was introduced. It takes place if the above conditions are both true, but in previous m frames the visual system mode was unstable. This warning mode is illustrated on the screen by the yellow frames of the video images.

This classification allows to coordinate the real-time functioning of system of aerial vehicle control and visual system as an additional source of current flight parameters. Here we suppose that the system of control can use the data from the visual system only if the last one is in the stable state. Parameter m (depth of previous frames, analyzed by the developed algorithm) was taken by us as m=1 for the current study and this choice is quite appropriate, but for further improvement of reliability of visual system functioning it can be increased to several sequential frames (m=2, 3 etc.). In addition, this classification can be applied for improvement the reliability of other sources of flight parameter like, for example, MCAS, discussed above, which was initially oriented on the use of data from only one sensor of angle of attack.

Testing of the multi-channel multi-threading algorithm for pitch and bank angles evaluating. Here we will test the multi-channel video system in dual– triple– and quad modes. For unification of input conditions, we use visual-content from the same video file, but, as noted above, with small contrast corrections for imitation of data from different cameras.

Dual-channel mode. The results for dual-channel mode of visual-data processing by the developed multi-threading algorithm are illustrated by Fig. (please see the Attachment, Fig. 3). In this case the achieved computational performance on the NVIDIA Jetson TX1 was equal to 4.8 fps vs 10 fps for a single-channel mode. So, twice increase of processing data size gives a little greater reduction of performance – in 2.083 times. Here 0.083 or 4.15% is the value of performance losses due to the use of multi-threading technology.

Triple-channel mode. The triple-channel mode of visual-data processing is demonstrated by Fig. (please see the Attachment, Fig. 4). The achieved computational performance on the NVIDIA Jetson TX1 for this case was equal to 3.18 fps. So, like in the previous dual-channel case the increase of processing data size in 3 times gives a little greater reduction of performance – in 3.145 times, where the multi-threading performance losses are equal 0.145 or 4.83%.

Quad-channel mode. The computational performance, achieved for the quad-channel mode on the NVIDIA Jetson TX1 was equal to 2.34 fps. Thus, the increase of processing data size in 4 times gives a little greater reduction of performance – in 4.274 times, where 0.274 or 6.84% – the multi-threading performance losses.

As it follows from the obtained results, multithreading technology works quite effectively on NVIDIA Jetson platform in multi-channel video processing, the losses of performance don’t exceed 7% for quad-channel and multi-channel video processing due to data redundancy allows to evaluate the reliability of the streaming information in real-time mode.

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