

Research Article

A New Concept of GPU-Accelerated On-Board Visual System for Aerial Vehicles

Yevhenii Shkvar*, Shi-ju E, Jian Cheng Cai, Zhechen Dai

Key Laboratory of Urban Rail Transit Intelligent Operation and Maintenance Technology & Equipment of Zhe Jiang Province, College of Engineering, Zhejiang Normal University, China
E-mail: shkvar.eugene@qq.com

Received: 2 June 2020; **Revised:** 20 June 2020; **Accepted:** 20 June 2020

Abstract: As technology advances, the intellectual ability and autonomy of the electronic-mechanical control system in modern aircrafts and spacecrafts continue to increase, which helps to reduce the crew load and accident rate. But these helpful controlling systems are not perfect. Under some unfavorable circumstances, they get stuck or start to function unpredictably when faced with a much more complicated real situation than the developers expected, even leading to crashes sometimes. The concept of visual multichannel processing support of aircraft/spacecraft launch and landing as an additional element of automatic control loop for flight safety and reliability improvement is proposed, and its advantages, feasibility, and expediency are discussed and evaluated. Visual analyzers are highly typical for the overwhelming majority of highly organized organisms (humans, animals, and insects) as the most informative source for controlling movement parameters. As a result, they have the potential to significantly enhance the reliability of the entire embedded vehicle control system. Moreover, their principal structure, implementation, and further functioning are very similar and universal across the actual flight operations of different vehicles. This opens up considerable prospects for their application in engineering, leveraging modern revolutionary advances in methodology and computational technologies for pattern recognition. In particular, it was shown that the required accuracy and productivity for real-world vehicle applications can be reached by developing the visual multichannel system as an additional source of flight state information. On the base of NVIDIA Jetson embedded portable low-power consuming Graphics Processing Unit (GPU)-accelerated massively parallel computational platform, providing Compute-Unified-Device-Architecture (CUDA) and Artificial Intelligence data processing in real-time mode.

Keywords: aircrafts, visual control, NVIDIA Jetson, embedded system, artificial intelligence

1. Introduction

Despite the steady increase in reliability and improvement of automation in the most advanced and high-tech aerospace fields, disasters of airplanes and spacecraft continue to occur with sufficient frequency even at present. Here are some recent disasters: the last two crashes of the latest revision of Boeing 737 Max-8 (October 29, 2018-Indonesia, March 10, 2019-Ethiopia) occurred due to the Maneuvering Characteristics Augmentation System (MCAS) failure; two An-148 aircrafts (March 5, 2011-Russia, February 11, 2018-Russia) crashed due to reaching outrageous speeds as a result of a malfunction of speed indicators, Israel's moon spacecraft Beresheet crashed to the moon surface (April

11, 2019) because of the inertial sensor and on-board computer failures; Japanese private rocket named MOMO-4 lost thrust due to onboard computer shutting down (July 27, 2019). Those disasters indicate that sometimes automatic systems are a major reason of piloting mistakes occurrence. Thus, the idea of sensors duplication and the need for redundant information on current parameters of vehicle movement is fruitful from the point of view of the reliability of any complicated object of control and the computing system redundancy is also relevant and they must be developed on the base of new effective computational architectures and modern computational platforms.

Our initial conceptual idea and initial proposal offer adding multiple (stereoscopic) 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. But this obvious at a first glance solution could not be implemented earlier [1]-[8], since it requires continuous realization of large volume of very productive on-board computations in real-time mode and with quite small energy consumption. The lack of the required computational on-board performance can be solved with the help of the wireless access of aerial vehicle to the remote computational system with required performance [1], but this approach is not quite reliable in comparison with any on-board solution. Modern computational systems, based on revolutionary massively-parallel technologies of computing, entirely correspond to this strong requirement. One of such implementations of massively-parallel computational technology is effectively developing since early 2007 by NVIDIA Corporation and now is well known as Compute-Unified-Device-Architecture (CUDA)-Compute Unified Device Architecture. The very effective combined Central Processing Unit (CPU) (Intel i7-8700K)-Graphics Processing Unit (GPU) (NVIDIA Quadro P5000) computational system for Unmanned Aerial Vehicles (UAV) pose estimating, described in [2], demonstrates GPU acceleration of the different algorithm elements by comparing with their only CPU-based efficiency in 7-38x times. So, this system demonstrates well the advantage of GPU acceleration of visual content recognition, but it was developed on the desktop computer components and oriented to its implementation in the Ground Control Station, so this is not an on-board integrated solution.

Latest hardware and software decisions of NVIDIA are concentrated to implementation of their 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. All the above-mentioned factors allow to consider NVIDIA Jetson platform as the perspective computational basis of the proposed additional multi-channel visual analyzer for aircrafts and spacecrafts. 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 Mobile Industry Processor Interface Camera Serial Interface (MIPI CSI) slots for in-built cameras. The most of existing schemes of visual support of navigation are based on a single-channel monocular vision [1], [2], [4]-[8], stereo vision approach, proposed and studied in [3], is dedicated for UAV detection as a target and in this case two video channels are needed for getting the distance to UAV, so, like in [1], [2], this is not an on-board solution.

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 (87 × 50 mm), has low level of power consumption (under 10 W) and supercomputing productivity (over 1 Tera Floating Point Operations Per Second (TFLOPs) computational performance, up to 1,400 Mpix/s camera interface capability) as well as equipped with a wide variety of traditional interfaces (Serial Advanced Technology Attachment (SATA) power + data, Full Secure Digital (SD), Universal Serial Bus (USB), Peripheral Component Interconnect (PCI) Express 2.0, M.2, High-Definition Multimedia Interface (HDMI), Gigabit Ethernet, WiFi, etc.) for different peripherals [9]. 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.

Development of such a concept of multi-channel visual-control system, based on NVIDIA model of massively parallel computations, could be very promising direction in designing the perspective vehicles and their controlling systems due to improvement of movement reliability and safety. In addition, such systems will be able to flexibly change flight parameters and effectively provide the stability control, especially for dynamically unstable vehicles that

can improve controllability and safety of flight as well as reduce fuel consumption. Moreover, adding the artificial intelligence function will make the controlling systems to be much more autonomous, which is very actual for increasing the reliability of spacecraft parameters control.

2. 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.

Any artificial real-time functioning multiple channel of visual content, as a very informative source of data for movement parameters control, requires high-performance computations and corresponding portable systems with small energy consumption. There are many quite effective embedded platforms for development portable computational systems (Arduino, Raspberry Pi, different Field-Programmable Gate Array (FPGA) systems, Paralela board and others), but the most promising among them for the formulated above problematics is the “NVIDIA Jetson” family of boards (TK1, TX1, TX2, Nano, Xavier), that realizes highly productive and scalable massively-parallel Graphics Processing Unit (GPU) accelerated model of computations. The analysis of effectiveness of one of this platform implementation (NVIDIA Jetson TX1) to the continuous real-time evaluations of flight parameters (pitch and bank angles) on the base of current visual information has been the subject of this research study.

The engineering actuality of the formulated proposal is flight safety and reliability improvement of automatically controlled launch and landing of aircrafts and spacecrafts with the help of NVIDIA AI embedded portable low power consuming massively-parallel computational Jetson platform.

3. The origins and background of the proposed concept of multiple video-channel for high-speed vehicles

Let’s firstly consider the visual analyzer propriety and actuality from the biomimetic point of view. Table 1 provides the comparative analysis of variety of different natural and artificial systems of control (in wide sense), that use visual-content processing.

Table 1. Comparative analysis of some natural and artificial visual analyzers

System name or owner	Advantages	Disadvantages
Nature (Organic)		
Jellyfishes’ primitive visual-based organic system of control (& Brainless, primitive sea animals: crustaceans, shrimps, mollusks, snails and insects: mosquitos, flies, ants etc.).	Have 24 floating eyes, able to sense the 3D space position, brainless, high neural processing is not required.	Low speed motion due to less neural processing, flexibility is pretty low.
Flying & Swimming biologics (fishes, birds, snakes etc.).	Wider vision (270° minimum, some can be upper to 360°) comparing with human (240°), high density of vision cells, able to sense object during high speed motion.	Wide vision lead to the lack of ability of sensing the object position and distance.
Human’s vision-motion system: Drivers.	Strong neural processing, able to process complicated tasks, natural cooperation between eyes and motion organic is pretty high, Highly spontaneous, able to deal with untrained tasks.	Vision range is low, eyes’ resolution is not high, great difference between different people, easily influenced by environment (brightness, weather etc.). It states of strong stress or in case of unexpected situation human’s brain aren’t able to generate the right decisions objectively.

Table 1. (cont.)

System name or owner	Advantages	Disadvantages
	Artificial (Engineering)	
Fly-by-Wire flight control system.	<p>Keeping the high-speed static stability at a relatively small value, which can significantly improve aircraft maneuverability, reduce trim resistance and aircraft agility.</p> <p>Associate with feedback control, the gain of the feedback channel and the main channel can be adjusted with the angle of attack, Mach number and dynamic pressure. This can meet the first-class quality requirements in the all-inclusive line regardless of the height and speed. This is the previous aircraft. What can't be achieved. It has the advantages of automatic trimming, automatic coordination of roll angle, boundary control, and improved warfare viability. The fly-by-wire flight control system has no mechanical system, is light in weight, small in size, and has no hysteresis caused by friction during operation, which can reduce maintenance. And can also improve aircraft performance through active control techniques such as gust load shedding, maneuvering load control, damping of the wing and fuselage structure and flutter suppression.</p>	<p>The reliability of the single-channel fly-by-wire system is not high enough. Due to the relationship between the quality of electronic components and design factors in a single-channel fly-by-wire system, the reliability of a single-channel fly-by-wire system is not high enough.</p> <p>The cost of the fly-by-wire system is high. In the case of a single-channel fly-by-wire system, the cost of the fly-by-wire system is lower than that of a mechanical control system. However, the fly-by-wire system must use a redundancy system to work reliably, so the overall cost of the fly-by-wire system is still relatively high. The fly-by-wire control system is susceptible to lightning strikes and electromagnetic pulse interference.</p>
Self-driving system: Autopilot (TESLA).	Hardware are equipped well, able to do lane keeping, steering, parking etc. Works fine on clear road.	Only focus on 2D motion, not able to process complicated road situation, system needs to be highly trained but cannot cover all cases (not all cars equipped with autopilot system, human-driving car moves unpredictably).

The overwhelming majority of biological objects from very small (insects) to very big and / or intellectual (human, elephant etc.) have a pair of eyes or even more (flies, spiders), also flying and swimming objects (birds, fish) have highly developed visual analyzers, which provide their owners stable, reliable and controllable movement. Moreover, the number of eyes of several insects is much more than only one pair (5-fly, 8 or even 12-spiders, 24-jellyfish “*Tripedalia cystophora*”) that provide them wide or even all-round visibility. Simple comparison of number of eyes and level of brain complexity of these insects allows to conclude that as the number of eyes is greater as the role of brain functionality in the visual data processing is smaller (in case of 24-eyed jellyfish brain is absent at all). This fact is positive from the point of view of feasibility of technical realization of video information in real-time mode, because it gives us the natural recipe-increasing the number of video sensors decreases the requirements to the productivity of computational device for data processing, which is always limited.

The second important aspect is associated with the number of basic eyes, that is for the most of biological organisms, including humans and animals, is even and is equal to two. Even insects, having odd number of eyes, have even (mostly two) basic highly-developed eyes and the rest are auxiliary, smaller and more primitive (for example, fly has two basic and 3 auxiliary eyes). This fact is a good background to consider two eyes as a reasonable number of channels for many different engineering applications, associated with vehicles movement, and its usefulness is not only associated with stereoscopic binocular vision properties, it can serve rather to all-round visibility. In addition, if we roughly try to associate the productivity of eye-brain video segment with some visual processing “frequency” it is for maneuverable insects like fly and dragonfly is about 200-300 Hz that is in 2-15 times faster in comparison with human eyes. It means, at least, the fact that the human’s video sensors are not quite perfect and fast even in comparison with insects, so their productivity in some very unfavorable circumstances can’t be enough in case of operating the complicated system like aircraft or another type of high-speed vehicle.

The mentioned above factors and, first of all, high level of visual channel informativeness has become the reason of engineering implementation of visual processing technology in the form of autonomous controlling systems like, for example, “AutoPilot”, designed by the one of the most successful electric-car company Tesla. Such functionality makes the Tesla car self-driving including lane keeping, steering and shifting lanes, recognizing the traffic lights with the help of the artificial visual analyzers, and this is the real demonstration of abilities and potential of the engineering embodiment of visual-based systems of control. One more bright demonstration of importance of computer vision and Artificial Intelligence (AI) technologies and visual data real-time smart processing for further aircraft development

is now actual Airbus's project "Autonomous Taxi, Take-Off & Landing" (ATTOL) [10], in frames of which during December 2019-January 2020 8 fully automatic vision-based take-offs of Airbus A350-1000 and reliable detection the runway and landing have been successfully realized.

The presented information allows to conclude that visual content analyzers are highly popular in the nature and their engineering implementations are promising. The success of the technical development of visual information processing allows to suppose about the usefulness and effectiveness of its application not only to an autonomous car, moving in limited two-dimensional space, but also to aircraft and spacecraft, having 3D trajectories and significantly higher speed.

4. Software and hardware aspects of single-channel visual content processing approach for aerial vehicle orientation recognition

The angular orientation is one of the important flying properties of the aerial vehicle for navigation and stability purposes. This is usually estimated through the integration of the vehicle's accelerations or angular rates, obtained by means gyroscopes. In contrast, vision-based systems can evaluate an aircraft's orientation directly with respect to the ground. The two degrees of freedom that are important for stability are the bank angle (Φ) and pitch angle (Θ). These two angles can be determined directly from the straight line representing the horizon of the image of the aircraft's forward-facing camera. The bank angle is easily determined as the arc tangent of the slope of the horizon slope. The pitch angle cannot be calculated accurately from the line of horizon, however the pitch angle is closely proportional to the percentage of image frame area above or below the line. The upper and lower percentages are represented by the symbols σ_a and σ_b respectively. We will focus on the use of σ_b because σ_a can be easily calculated from σ_b ($\sigma_a = 1 - \sigma_b$). In a rectangular image, the relationship between σ_b and pitch angle is non-linear and can be slightly couple slightly with Φ and may vary with camera distortion. The difference can be resolved using a calibration table, however Scott M. Ettinger's flight tests have shown satisfactory performance using a linear relationship between test and σ_b [11]. Scott M. Ettinger's research team's new algorithm makes no prior assumptions about the image data (sky and ground strength) [12]. The only real assumption of this algorithm is that horizontal lines can be approximated as straight lines. The new technique is partially derived from an introspective analysis of the way humans may approach this problem. Obviously, humans do not try to fit the step function to every vertical line in the image. We may use some form of edge detection, but usually with knowledge of the physical world, look for two different adjacent regions in the image, each area has its own color and texture attributes. For the purpose of this work, it is desirable to choose or develop such algorithms that can run in real-time on embedded computational system due to effective usage the GPU-accelerated technology. Therefore, obtaining the texture information requires a lot of calculation, but the information about color can be obtained immediately from the image data, that reduces the complexity of the algorithm by limiting the use of color information and, as a result, significantly reduces its needs in computational resources and improves the functioning efficiency. We understand well about existence of other different approaches and more precise and complicated algorithms, 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 UAV "Autonomous Cargo Aircraft" [13], [14], 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.

In this study, we have been using GPU-based NVIDIA Jetson TX1 developer platform. Professor of University of Tennessee, author of "Linpack", Jack Dongarra said: "GPUs have evolved to the point where many real-world applications are easily implemented on them and run significantly faster than on multi-core systems. Future computing

architectures will be hybrid systems with parallel-core GPUs working in tandem with multi-core CPUs". The main reason why use GPU is the GPU has evolved into a very flexible and powerful processor, it's programmable using high-level languages, it supports 32-bit and 64-bit floating point IEEE-754 precision, it offers lots of GFLOPS and GPU is in every PC and workstation. The GPU is specialized for compute-intensive, highly parallel computation (exactly what graphics rendering is about), compare the core number with CPU, GPU has much more transistors can be devoted to data processing rather than data caching and flow control and is able to handle larger data sets and massive data parallelism. CUDA parallel programming architecture allows to effectively compose a program mainly for one thread and execute it by many GPU parallel threads on any number of NVIDIA GPU cores by means special instructions without recompiling.

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. While OpenCV itself doesn't play a critical role in deep learning, it is used by other deep learning libraries such as Caffe, specifically in "utility" programs (such as building a dataset of images). Simply put, having OpenCV installed makes it easier to write code to facilitate the procedure of preprocessing 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.

5. Results of testing the single-channel horizon-detecting algorithm implementation

This step of study is dedicated to analysis of efficiency of processing the basic single-channel video content from one camera (or other alternative source of input data like video-or image-files). Running the Python code in its original version [13] on the NVIDIA Jetson TX1 platform with different input data, we've got the following results.

Static images recognition. Figure 1 shows the results of the chosen algorithm for several typical videos frames. Here, for the quality image of clear horizon line-the simplest and the most favorable case (Figure 1a) the algorithm is able to find it correctly. Even if the video signal in this image is severely corrupted by video transmission noise, the algorithm can still find the correct horizontal line. The algorithm is still able to find the line closest to the actual horizontal line.

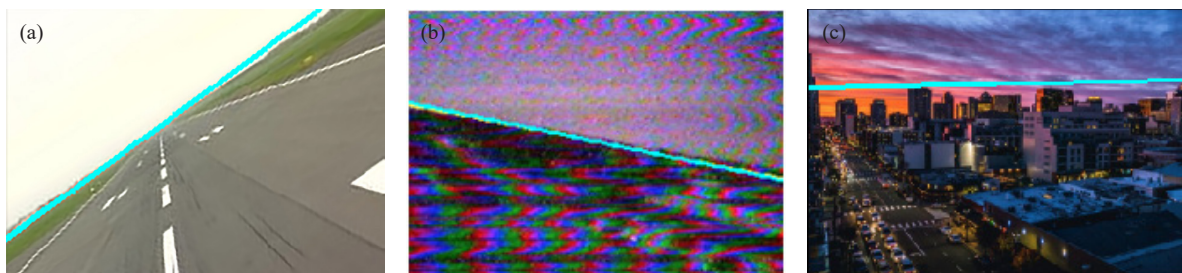


Figure 1. Horizon-detection: (a) Quality image; (b) Image heavily distorted noise; (c) Non-linear skyline image

Video-content recognition. Then we tested a video downloaded from YouTube.com which has the content taken from the camera that mounted on the front of the aircraft, since we're not able to put this system on a real air vehicle. The results show that it has the ability to successfully detect the horizon line in a real-time mode and there is nearly no time lag. And also, is able to handle noisy video information and nonlinear edge detection. The testing video detective is illustrated by Figure 2, showing various horizon-detection cases under different lighting conditions, and with varying degrees of video transmission noise. For each example, the cyan-blue line indicates the algorithm's horizon estimate. Also, on the up-left corner, the system successfully calculates the bank angle with the information of proportion of the image above and below.

When the video test starts running, a significant increase in GPU load up to 60% can be observed from the system monitor, while the average CPU load increases by only about 25%. As shown in Figure 3, the Advanced RISC Machine (ARM) cores primarily support the GPU-accelerated computation process, indicating that the GPU is utilized for data

processing due to CUDA support. The test was run on a powerful CPU chip-the Intel i5-5257U with 2 cores and 2.3 GHz- and similarly on the NVIDIA Jetson TX1 with CUDA support for 10 times. The execution time of each test run was recorded, and the results were averaged. These average results (5.6 frames per second (fps) for NVIDIA Jetson TX1 and 3.9 fps for Intel i5-5257U) demonstrated the real GPU acceleration of the computations about 1.44 times in comparison with CPU. 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 on the base of considering algorithm. Its further generalization for the case of independent multi-channel video-content will be described and discussed later.

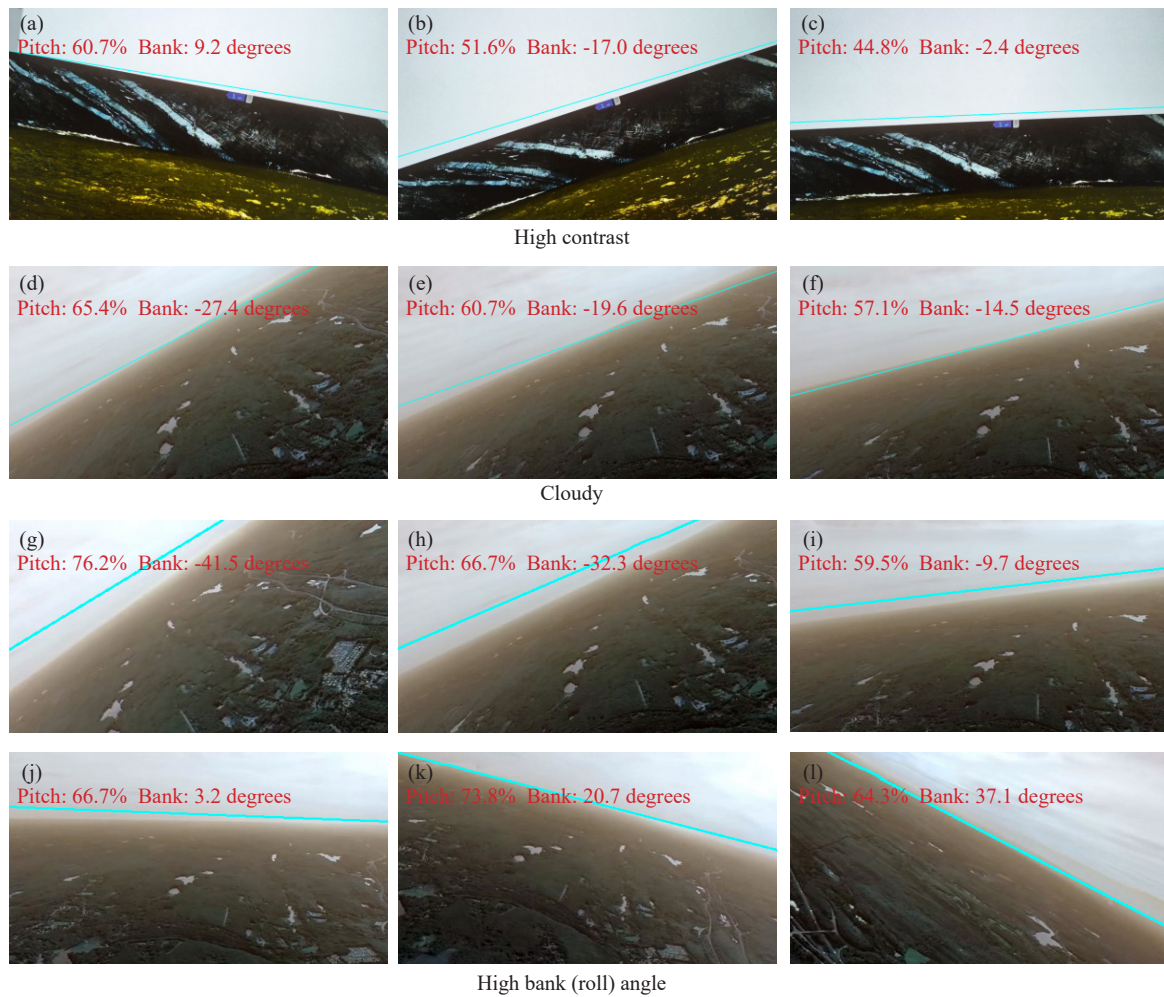


Figure 2. Various horizon-detection examples

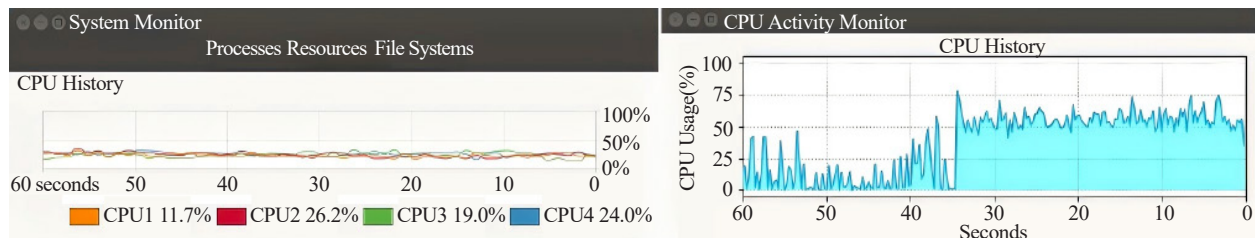


Figure 3. Real-time CPU and GPU usage

6. Results of testing the multi-channel horizon-detecting algorithm implementation

The last step of this study is associated with the initially formulated goal-expand the tested above algorithm for the case of independent multi-channel horizon-detecting visual processing. Technologically, 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). There is only one real obstacle here-limited resources of Random Access Memory (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 performance redistribution evaluation 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).

6.1 An error estimation algorithm and matching measurements of pitch and bank angles

6.1.1 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 ($i = 0, \dots, N$), 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.

(1) 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 $\varepsilon_{\max} = 2^\circ$. 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 (Figure 4).

(2) 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 (Figure 5).

(3) 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 (Figure 6).

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, Maneuvering Characteristics Augmentation System (MCAS), noted above, which was initially oriented on the use of data from only one of two existing sensors of angle of attack.

6.2 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 Figure 4. 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.

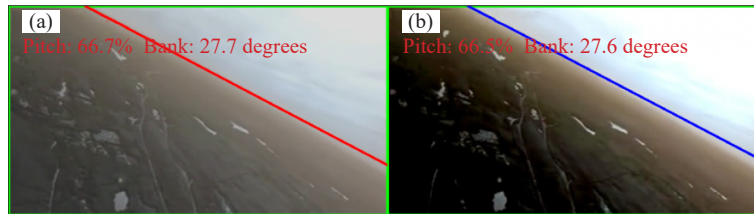


Figure 4. Dual-channel parallel visual content GPU-accelerated processing

Triple-channel mode. The triple-channel mode of visual-data processing is demonstrated by Figure 5. 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%.

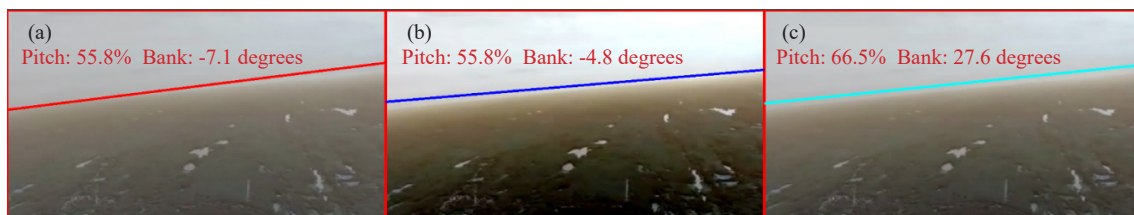


Figure 5. Triple-channel parallel visual data GPU-accelerated processing

Quad-channel mode. Visual-data processing by the quad-channel mode is represented by Figure 6. The computational performance, achieved for this case 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.

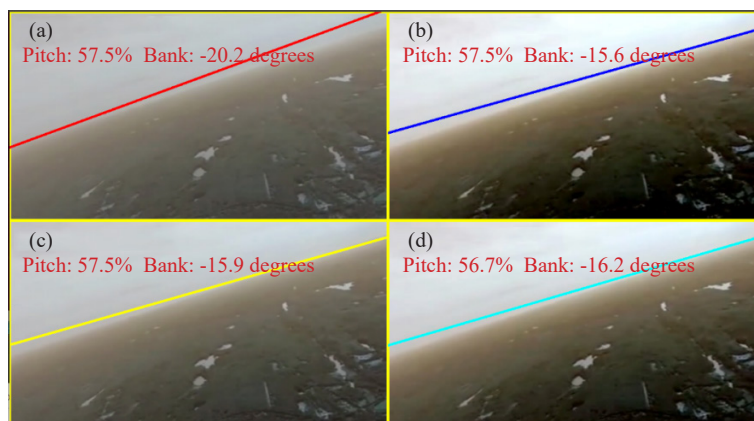


Figure 6. Quad-channel parallel visual data GPU-accelerated processing

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. In addition, multi-channel video processing due to data redundancy allows to evaluate the reliability of the current information in real-time mode.

7. Conclusions

1. The multi-channel vision-based and GPU-accelerated concept of improvement of control systems of aircraft flight stability has been proposed.

2. The critical analysis of the disadvantages of existing and recently introduced principles of flight control has been made.

3. The analysis of natural visual analyzers for different biological creatures, including humans, animals, insects, etc. has been provided and the conclusion about its universality and information efficiency has been done.

4. The methodology of multi-channel visual-based channel implementation into flying vehicles has been proposed.

5. The requirements to the vision-based control system have been worked out (real-time functioning, reliability, massively parallel data processing, etc.) and the conclusion about expediency to build the visual channel of the control system on the base of the GPU -accelerated computational platform has been worked out.

6. The methodology of the images (horizon) recognition (artificial intelligence theory and its practical implementations), supercomputing computations in real-time mode (methods of Graphics Processing Unit programming on the base of the Compute-Unifed-Device-Architecture (CUDA) platform) and their implementations in the embedded portable energy-consuming systems (NVIDIA Jetson platform) have been elaborated.

7. The algorithm of the horizon detection has been improved for the multi-channel and multithreading case and tested.

8. The corresponding software environment, OpenCV library, compiled with CUDA, has been modified and improved for its effective use on the embedded NVIDIA Jetson TX1 computational platform and the obtained results have demonstrated the efficiency of GPU-based computations in comparison with CPU-based embedded systems (Intel 2.3 GHz, 2 Cores i5) in 1.5 times.

9. Future research in this field will be oriented to the following actual directions:

(1) We will explore other optimization standards and features, such as textures, to extend our current approach to monochrome imaging sensors. Finally, we are considering more general techniques for flight control, based on the parallel analysis of multi-channel video-content from several cameras to take advantage of GPU implementations for simultaneous processing the threads of graphical information in real-time mode.

(2) Developing the conceptual scheme of the methodology of improving the reliability of the control system of aerospace complicated vehicles on the base of onboard simultaneous processing the video content in real-time mode based on CUDA computational architecture and supercomputing power of embedded portable NVIDIA Jetson platform is actual, feasible and promising. And its realization can give a significant social benefit, because of allowing to substantially improve the reliability of existing and prospective vehicles on the base of conceptually new real-time functioning embedded intellectual systems of flight control. It should be recommended to further development and practical implementation.

Acknowledgement

The authors team is gratefully thankful to NVIDIA Corporation for support and the donation of the Jetson TX1 developer kit, used for this research.

Conflicts of interest

The authors declare no conflict of interest.

References

- [1] S. Timotheatos, S. Piperakis, A. Argyros, and P. Trahanias, "Vision based horizon detection for UAV navigation," in Proceedings of the 27th International Conference on Robotics in Alpe-Adria Danube Region (RAAD 2018), 2018, pp. 181-189.
- [2] N. P. Santos, V. Lobo, and A. Bernardino, "3D model-based UAV pose estimation using GPU," in Oceans 2019 MTS/IEEE Seattle, 2019, pp. 1-6.
- [3] M. N. Brunet, G. A. Ribeiro, N. Mahmoudian, and M. Rastgaar, "Stereo vision for unmanned aerial vehicle detection, tracking, and motion control," *arXiv:2005.04183*, 2020. [Online serial]. Available: <https://doi.org/10.48550/arXiv.2005.04183>. [Accessed April 23, 2020].
- [4] M. Petković, I. Vujović, and I. Kuzmanić, "An overview on horizon detection methods in maritime video surveillance," *Transactions on Maritime Science*, vol. 9, no. 1, pp. 106-112, 2020.
- [5] J. Rademeyer, "Vision-based flight control for a quadrotor UAV," M. S. thesis, Stellenbosch University, Stellenbosch, South Africa, 2020.
- [6] C. Y. Jeong, H. S. Yang, and K. D. Moon, "Horizon detection in maritime images using scene parsing network," *Electronics Letters*, vol. 54, no. 12, pp. 760-762, 2018.
- [7] P. Hecker, M. Angermann, U. Bestmann, A. Dekiert, and S. Wolkow, "Optical aircraft positioning for monitoring of the integrated navigation system during landing approach," *Gyroscopy and Navigation*, vol. 10, pp. 216-230, 2019.
- [8] O. V. Zbrutsky, J. O. Malysheva, and V. V. Burnashev, "Accuracy of integrated aircraft navigation system with optical horizon sensor," *Information Systems, Mechanics and Control, NTUU "KPI"*, no. 11, pp. 5-12, 2014.
- [9] NVIDIA, "NVIDIA Jetson TX1 Module: Technical Specifications," 2015. [Online]. Available: <https://images.nvidia.com/content/tegra/embedded-systems/pdf/JTX1-Module-Product-sheet.pdf>. [Accessed April 23, 2020].
- [10] Airbus, "Is Autonomy the Future of Aerial Mobility?" *Airbus Newsroom*, Jan. 16, 2020. [Online]. Available: <https://www.airbus.com/en/newsroom/stories/2020-01-is-autonomy-the-future-of-aerial-mobility>. [Accessed April 23, 2020].
- [11] S. M. Ettinger, "Design and implementation of autonomous vision-guided micro air vehicles," M.S. thesis, Electrical & Computer Engineering, University of Florida, 2001.
- [12] S. M. Ettinger, M. C. Nechyba, P. G. Ifju, and M. Waszak, "Vision-guided flight stability and control for micro air vehicles," *Advanced Robotics*, vol. 17, no. 7, pp. 617-640, 2003.
- [13] N. S. Bradford, HorizonCV: Horizon Detection for UAVs with Python. GitHub Repository. 2016-2017. [Online]. Available: <https://github.com/nsbradford/HorizonCV>. [Accessed April 23, 2020].
- [14] N. Cyganski, B. Gillette, K. Henson, K. Xu, and E. Nadel, "The autonomous cargo aircraft project ACAP," Major Qualifying Project, Worcester Polytechnic Institute, Worcester, MA, 2017.