Multi-version approach to improve the reliability of processing data of the earth remote sensing in the real-time

Mikhail V. Saramud[1], *Igor V.* Kovalev[1,2], *Vasiliy V.* Losev[1], *Mariam O.* Petrosyan[1], *and Dmitriy I.* Kovalev[2]

[1] Reshetnev Siberian State University of Science and Technology 31, Krasnoyarsky Rabochy Av.,Russia, 660037
[2] Federal State Autonomous Educational Institution of Higher Education Siberian Federal University, 79, Svobodny Pr., Krasnoyarsk, Russia, 660041

Abstract. The article describes the use of a multi-version approach to improve the accuracy of the classification of images when solving the problem of image analysis for Earth remote sensing. The implementation of this approach makes it possible to reduce the classification error and, consequently, to increase the reliability of processing remote sensing data. A practical study was carried out in a multi-version real-time execution environment, which makes it possible to organize image processing on board of an unmanned vehicle. The results confirm the effectiveness of the proposed approach.

Krasnoyarsk region is a vast territory, which includes forests, requiring constant monitoring and control [1]. There is a need to quickly identify forest fires, pest infestations, illegal logging and other situations that may pose a danger to people and nature [2]. Earth remote sensing (ERS) allows solving this problem quickly, as well as conducting research, creating geographic information systems that are necessary for enterprises to solve production problems - from laying cable networks to the development of mineral deposits.

With Earth remote sensing, the task of automated processing of a huge amount of information arises. It is not possible to process such volumes manually using operators. Investigation of Earth remote sensing images by man is only possible for a specific area of interest due to its identification by an automated system or for other reasons. Modern image processing algorithms and decision-making algorithms based on them are not perfect. As a rule, the automated study of large areas is the task of dividing regions into given classes: “norm”, “fire”, “struck by a pest”, “cutting down”, etc [3]. However, classification errors are not excluded. In case of an erroneous classification of a normal area as a problem, for example, a “fire”, this will only lead to a waste of time for manual checking and additional research of this area. The case of the erroneous classification of the problem area as normal can lead to disastrous consequences. Thus, the problem arises of increasing the accuracy of the classification when processing images of Earth remote sensing.

One of the most effective known ways to improve the quality of software systems is a multi-version approach. To be able to use it, we need several different, but functionally equivalent software “versions” [4]. In our case, the versions correspond to different classification algorithms. In this task, multi-version can be easily achieved, since today there are many different algorithms of this class.

Moreover, there are often several sources of information about the same territory. These can be images in different ranges (infrared, ultraviolet, radio spectrum of the electromagnetic spectrum, etc.), from different satellites and aircrafts. In this case, you can create a multi-version pool of software implementations of a single algorithm that decides on various sources of information.

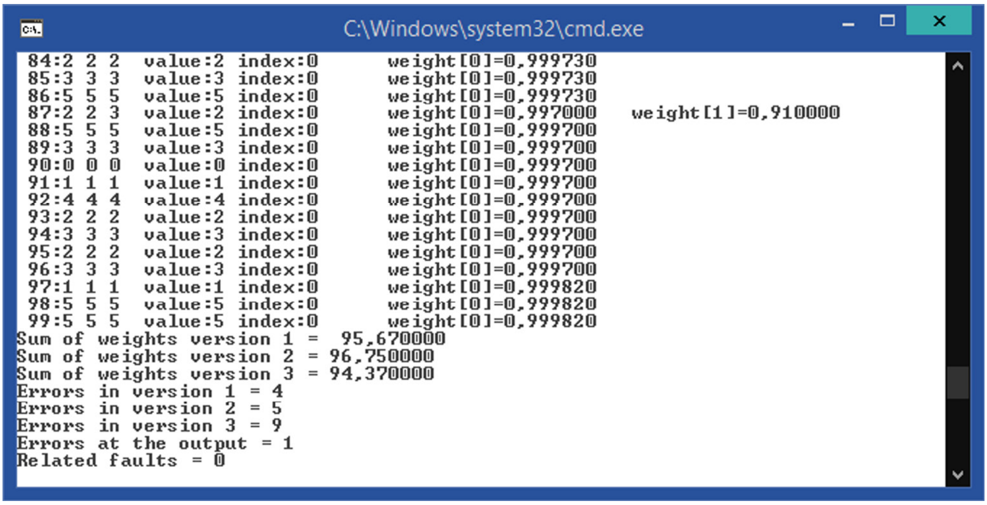
As practice shows, multi-version systems significantly increase both the reliability and the quality of decision making in the system. For practical testing of this approach in a real multi-version execution environment, three verification algorithms are implemented, each of which decides whether the same area belongs to one of the 6 specified classes. Based on the outputs of these algorithms, the system will make the final decision.

The software implementation of the environment in which the verification algorithms are executed is a multi-version real-time execution environment. This environment is based on the real-time operating system FreeRTOS v10 and is a set of C source files for compilation [5]. This allows you to make modifications and extensions to the code of the execution environment itself and to develop new modules.

The execution environment itself has rather low hardware requirements and allows assembly for most modern single-chip platforms. This fact allows organizing image processing directly "on board" of an autonomous air vehicle or space vehicle. In such conditions, the bottleneck is the communication channel, which does not allow transmitting a constant stream of images for processing to the ground. The ability to process, and most importantly, make decisions on board and transfer to earth only the decision made and only the image of the required area of interest, greatly simplifies the process of Earth remote sensing in order to quickly identify problem areas.

As a decision-making algorithm in multi-version systems in this software implementation, the author's modification of the weighted voting algorithm by a coordinated majority with the forgetting element is used [6]. The version weights provided for by this algorithm (necessary for the operation of the algorithm) are, in their sense, an assessment of the quality of the version, and therefore the algorithm it implements. The presence of version weights will not only increase the reliability at the decision-making stage, but also provide estimates of the quality of the algorithms used. Thus, we will receive additional information on the operation of the algorithms in real conditions of the execution environment.

Figure 1 shows the results of the operation of testing algorithms in a multi-version execution environment. Considering the results obtained, we note that the checking algorithms made 4, 5 and 9 errors, respectively.



```
 84:2 2 2  value:2 index:0      weight[0]=0,999730
 85:3 3 3  value:3 index:0      weight[0]=0,999730
 86:5 5 5  value:5 index:0      weight[0]=0,999730
 87:2 2 3  value:2 index:0      weight[0]=0,997000   weight[1]=0,910000
 88:5 5 5  value:5 index:0      weight[0]=0,999700
 89:3 3 3  value:3 index:0      weight[0]=0,999700
 90:0 0 0  value:0 index:0      weight[0]=0,999700
 91:1 1 1  value:1 index:0      weight[0]=0,999700
 92:4 4 4  value:4 index:0      weight[0]=0,999700
 93:2 2 2  value:2 index:0      weight[0]=0,999700
 94:3 3 3  value:3 index:0      weight[0]=0,999700
 95:2 2 2  value:2 index:0      weight[0]=0,999700
 96:3 3 3  value:3 index:0      weight[0]=0,999700
 97:1 1 1  value:1 index:0      weight[0]=0,999820
 98:5 5 5  value:5 index:0      weight[0]=0,999820
 99:5 5 5  value:5 index:0      weight[0]=0,999820
Sum of weights version 1 =  95,670000
Sum of weights version 2 = 96,750000
Sum of weights version 3 = 94,370000
Errors in version 1 = 4
Errors in version 2 = 5
Errors in version 3 = 9
Errors at the output = 1
Related faults = 0
```

Fig. 1. The results of the work of verification algorithms in the execution environment.

Based on the responses of these algorithms, the multi-version system always chose the right answer, making only one mistake. That is, the system made four times less classification mistakes than the best algorithm. This proves the effectiveness of the multi-version approach, since the quality of the system operation with its application exceeds the quality of operation of any of the algorithms used in it.

Conclusion

The obtained results prove the applicability of the multi-version approach for solving the problem of increasing the reliability of processing remote sensing data and show the high efficiency of this approach. The use of multi-version models in data processing will increase the reliability of the results obtained and reduce the classification error.

The considered practical solution based on the real-time operating system FreeRTOS is fault tolerant, not demanding on the resources of the hardware platform and portable. This allows organizing a streaming data processing system, even within an unmanned vehicle. This is a very important factor, since it is extremely difficult to transmit a stream of images via communication channels during the flight, and if they are processed “on board,” the device will already report the result - the identified problem in a certain area.

Such an opportunity will significantly speed up the detection of forest fires and other events requiring prompt response.

Acknowledgments

This work was supported by Ministry of Education and Science of Russian Federation within limits of state contract № 2.2867.2017/4.6.

References

1. Korobko A.V., Penkova T.G., Nicheporchuk V.V., Mihalev A.S. The integral OLAP-model of the emergency risk estimation in the case of Krasnoyarsk region // 2013 36th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp. 1161 – 1166 (2013)

2. Quan Zou, Guoqing Li, Wenyang Yu An integrated disaster rapid cloud service platform using remote sensing data // 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 5221 – 5224 (2017)
3. Yuhuan Ren, Yalan Liu Geological disaster detection from remote sensing image based on experts' knowledge and image features // 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 677 – 680 (2016)
4. Kovalev I. V., Losev V. V., Saramud M.V., Kovalev D.I., Petrosyan M.O. To the question of implementation of multi-version execution environment software of onboard autonomous pilotless objects by means of real-time operating system // Vestnik SibGAU, 2017. Vol 18, №1, p. 58 – 61. (2017)
5. Saramud M.V., Kovalev I. V., Losev V. V., Petrosyan M.O. Application of FreeRTOS for implementation of the execution environment of real-time multi-version software // International Journal on Information Technologies & Security, No 3 (vol. 10), pp. 75 – 82. (2018)
6. Igor Kovalev, Anna Voroshilova, Vasiliy Losev, Mikhail Saramud, Maria Chuvashova, Aleksandr Medvedev. Comparative Tests of Decision Making Algorithms for a Multiversion Execution Environment of the Fault Tolerance Software // Proceedings of 2017 European Conference on Electrical Engineering and Computer Science (EECS 2017), pp 211 – 217 (2017)