UDC 004

Information, Computing and Intelligent systems. – 2021. – № 2. – P. 83.

Recommended for publication by the Academic Council of the Faculty of Computer Science and Computer Engineering

The scientific journal “Information, Computing and Intelligent systems” is intended for the publication of the results of scientific research and scientific and practical developments in the field of technical sciences by students, masters, PhDstudents, scientists, and practicing specialists in the field of science "Information systems".

The thematic orientation of the journal “Information, computing and intelligent systems” is reflected in the following headings: computerized and computer systems and networks, information technologies, the Internet of Things, information transformation and processing, cloud computing, computer cryptography, data protection, intelligent systems, artificial intelligence, machine learning, automated design of software and technical tools, system control, diagnostics and control of parameters of complex systems, processes and environments; engineering knowledge, embedded systems, robotics, microelectronics.

ISSN 2708-4930

Certificate of state registration No. 23827-13667ПР from 20.02.2019
Magazine in English
Web- resource – http://itvisnyk.kpi.ua/
Format 60×84 1/8. Garnitura Times. Offset Folder № 1.

Ministry of Education and Science of Ukraine
National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”

Information, Computing and Intelligent systems

SCIENTIFIC EDITION

The journal is the legal successor of the Collection of scientific works
“Bulletin of NTUU “KPI”. Informatics, control and computer engineering”

Founded in 1964 years

Issue 2

Kyiv – 2021 (2)

SUMMARY

UDK: 004.77, 004.728
DOI: https://doi.org/10.20535/2708-4930.2.2021.244169

V. HER
V. TARANIUK
V. TKACHENKO
I. KLYMENKO
S. NIKOLSKYI

METHODOLOGY OF NETWORK ENVIRONMENT TESTING FOR IoT DEVICES

The article reviews methods and technologies for testing the network environment of embedded systems and writing test documentation. As an example, a testing technique based on a defect report has been developed. A performance test was developed to check the load of the embedded device's network environment using special bash scripts for performance testing.

Keywords: IoT, embedded system, test case, defect report, troubleshooting, performance testing.

1. Introduction

1.1 What is the Internet of Things and IoT Testing?

The Internet of Things, or IoT, is a term for billions of devices connected to the Internet. Each of these devices collects data and exchanges it with other devices. The Internet of Things phenomenon is due to the presence of powerful network bandwidth, the proliferation of wireless networks and low-cost computer chips [1-3].

The Internet of Things connects various objects, adds modern sensors and facilitates data exchange between different devices in real time [4]. This allows people to add a new level of digital intelligence to their devices and help them process information locally without any delay.

However, traditional software testing does not work for the Internet of Things [5]. Testers should pay attention to user-centric testing and prevent errors, not detect them. This means that quality assurance engineers have a role to play both in operation and in development.

To guarantee the quality of the software, testers need to gain in-depth knowledge in the subject area. Testers who have no experience testing embedded systems or equipment should develop their skills in these areas.

1.2 Importance of IoT Testing

IoT is data exchange and collaboration in real time [3]. Performance issues in any part can negatively affect the performance of another network. One node compromised as a result of a cyber attack can harm others.

Reliable Internet of Things testing ensures system predictability and prevents unexpected failures. You can easily identify weak network nodes in advance and take appropriate measures to increase system reliability. This will provide the end user with a flawless customer experience.

Vulnerabilities in IoT devices will help you determine the reliability of data privacy and digital security. In the absence of reliable cybersecurity, hackers can change the process or falsify data by gaining control of the IoT network. Therefore, before each update, you must thoroughly test the IoT device.

In addition, the fragmented nature of the Internet of Things ecosystem makes testing even more difficult. To do this, you need to have large and reliable test teams equipped with multiple platforms and devices to ensure greater compatibility between different channels [6, 7].

In addition, to ensure that IoT applications work as expected, several other factors need to be considered:

- Ensure that your IoT devices are securely connected to sensors, the cloud, other IoT devices, and other elements needed to ensure unified interaction.
- Make sure the Internet of Things is able to ensure continuity

- Must meet international standards
- Harmonious and flawless work with other interconnected devices of the Internet of Things
- Data obtained from the Internet of Things must be protected from malware and other security vulnerabilities.

Reliable testing of IoT devices is the only effective way to effectively address the above factors. The general approach to quality assurance when testing IoT devices is to reduce testing periods to launch a reliable product with faster market launch. Therefore, it is important to conduct testing from the beginning of the development phase to identify and correct deficiencies at an early stage.

Problem statement: It is necessary to test IoT devices in order to avoid improper operation, which can lead to fatal consequences (such as security problems, communication problems, failure of sensors or the device itself, etc.). To prevent such problems, it is necessary to thoroughly test the IoT devices and their network environment before use.

2. Review of existing solutions and their use

Embedded systems are used in a wide range of technologies in various industries [2, 3]. Some examples:

Cars. Modern machines usually consist of many computers (sometimes up to 100) or embedded systems designed to perform various tasks in the vehicle. Some of these systems perform basic utility functions, while others provide entertaining or user-oriented functions. Some embedded systems in consumer cars include cruise control, backup sensors, suspension control, navigation systems and airbag systems.

Mobile Phones. They consist of many embedded systems, including graphical user interface software and hardware, operating systems (OS), cameras, microphones, and USB I / O modules (universal serial bus).

Industrial machines. They may contain embedded systems, such as sensors, and may themselves be embedded systems. Industrial machines often have built-in automation systems that perform specific control and management functions.

Medical equipment. They may include embedded systems such as sensors and controls. Medical equipment, like industrial machines, must also be very user-friendly so that machine errors that can be prevented do not endanger human health. This means that they will often include more complex operating systems and a graphical interface designed for the corresponding interface.

There are many examples of embedded systems. So, the article will show several examples and analyze their use, as well as their structure.

Automotive Embedded Systems:

Electronic control units are used in automotive embedded systems. This unit contains a microcontroller, switches, sensors, drivers, etc. All sensors and actuators are connected to the electronic control unit. Cars that use embedded systems can consist of hundreds of microprocessors. Each microcontroller performs its own special task. Some of them control the engine. Some launch the dashboard device. The whole system actually consists of several small systems. The use of embedded systems in the automotive industry has reduced the cost factor. This has improved overall performance and increased functionality. It has also reduced weight and made cars safer and more reliable. Applications of automotive embedded systems include [5]:

- Automatic stability control
- Traction control system
- Pre-emergency safety system
- Airbag
- Car navigation system

So you can see that embedded systems have improved cars and made them more comfortable. Also, they have increased the functionality of cars and made them easy to use.

Home Security System:

Home security systems are widely used today [1]. These systems have several functions, such as checking for fire or gas leaks, as well as detecting attempts by a suspicious person to enter the house. The microcontroller is used to control all operations. Sensors give data and if something wrong happens then safety alarms get activated. Sensors used in such systems include gas sensors, smoke sensors, temperature sensors, IR sensors, etc. Such systems also include a keyboard for entering passwords on the gate. If the correct password is entered, this built-in system opens the gate, and if someone tries to enter the wrong password, the alarm goes off and the gate remains closed. The output signal comes from alarms or any display. The conclusion can also be sent to a remote location. If family members are not at home, they can still monitor what is happening in their home. The home security system is not limited to homes. Such systems can be used in stores, shops and in production. Almost every industry and office has security systems that can recognize workers by their faces or IDs. The home automation system is also one example of embedded systems as a home security system.

3. What is an embedded system?

3.1 Embedded systems

An embedded system is a computer system with a specific function in a larger mechanical or electrical system. They control many commonly used devices. They consume little energy, are small in size and their cost is low per unit. Modern embedded systems are often based on microcontrollers. A microcontroller is a small computer on a single integrated circuit that contains a processor core, memory, and programmable I/O peripherals. Because the embedded system is designed to perform certain tasks, they can be optimized to reduce the size and cost of the product, as well as increase reliability and performance [6].

Embedded systems have revolutionized science. It is also part of the Internet of Things (IoT), a technology in which objects, animals, or people are given unique identifiers and the ability to transmit data over a network without requiring human-to-human or human-to-computer interaction.

3.2 How embedded systems work

Embedded systems always function as part of a single device - this is what is meant by the term "embedded". These are inexpensive, low-power, small computers built into other mechanical or electrical systems. They typically consist of a processor, power supply, memory, and communication ports. Embedded systems use communication ports to transfer data between the processor and peripherals - often other embedded systems - using a communication protocol. The processor interprets this data using minimal software stored in memory. Software is usually highly dependent on the function performed by the embedded system. The processor can be a microprocessor or a microcontroller. Microcontrollers are just microprocessors with peripheral interfaces and built-in memory. Microprocessors use separate integrated circuits for memory and peripherals instead of including them in the chip. Both can be used, but microprocessors usually require more support circuits than microcontrollers because they are less integrated into the microprocessor.

3.3 The structure of embedded systems

Embedded systems vary in complexity, but usually consist of three main elements:

- Hardware. The hardware of embedded systems is based on microprocessors and microcontrollers. Microprocessors are very similar to microcontrollers and usually refer to a CPU (CPU) that is integrated with other basic computing components, such as memory chips and digital signal processors (DSP). These components are built into the microcontrollers on a single chip.
- Software and firmware. Embedded software software can vary in complexity. However, industrial-grade microcontrollers and embedded IoT systems typically run very simple software that requires a small amount of memory.
- Real time operating system. They do not always work in embedded systems, especially in smaller systems. RTOS determine how the system works by controlling the software and setting rules during program execution.

In terms of hardware, a basic embedded system would consist of the following elements:

- Sensors convert physical sense data into an electrical signal.

- Analog-to-digital (A-D) converters change an analog electrical signal into a digital one.
- Processors process digital signals and store them in memory.
- Digital-to-analog (D-A) converters change the digital data from the processor into analog data.
- Actuators compare actual output to memory-stored output and choose the correct one.

The sensor reads external inputs, the converters make that input readable to the processor, and the processor turns that information into useful output for the embedded system.

3.4 Types of embedded systems

There are several basic types of embedded systems that differ in their functional requirements. Among them are:

- ***Mobile embedded systems*** are small systems designed for easy use and everyday portability. An example of this is digital cameras.
- ***Network-embedded systems*** are connected to the network to provide output to other systems. Examples are home security and point of sale (POS) systems.
- ***Standalone embedded systems*** do not depend on the main system. Like any embedded system, they perform a specialized task. However, they do not necessarily belong to the host system, unlike other embedded systems. An example of this is a calculator or MP3 player.
- ***Embedded systems in real time*** give the desired result after a certain time interval. They are often used in the medical, industrial and military sectors because they are responsible for the most important tasks that depend on time. An example of this is the traffic control system.

Embedded systems can also be categorized by their performance requirements:

- ***Small-scale embedded systems*** often use no more than an 8-bit microcontroller.
- ***Medium-scale embedded systems*** use a larger microcontroller (16-32 bit) and often link microcontrollers together.
- ***Sophisticated-scale embedded systems*** often use several algorithms that result in software and hardware complexities and may require more complex software, a configurable processor and / or a programmable logic array.

There are several common architectures of embedded system programs that become necessary as embedded systems grow and become more complex in scale. They include:

- ***Simple control cycles*** caused by routines that control a specific piece of equipment or firmware.
- ***Interrupt control systems*** have two cycles: primary and secondary. Interrupts in cycles cause tasks.
- ***Cooperative multitasking*** is essentially a simple control loop located in the Application Programming Interface (API).
- ***Warning multitasking or multithreading*** is often used with real-time operating systems (RTOS) and has strategies for synchronizing and switching tasks.

To sum up: Today, any embedded system is an IoT device that connects to the Internet and can communicate with other IoT devices. Also, the proposed complex allows you to test the network environment of the embedded system.

4. Types of test documentation and methodology

4.1 Test documentation

The basis of testing is knowledge and ability to write test documentation, as well as knowledge of different testing methodologies. There are three main types of documentation: test case, defect report, and troubleshooting. There are also different approaches and testing methods. Examples based on the TCP protocol stack will be shown. Let's take a closer look at the test documentation, which is aimed at helping developers and testers improve the product being developed.

Test case: A test case is a set of actions performed on a system to determine whether it meets the requirements of the software and is functioning properly [8]. The purpose of the test case is to determine whether the various functions in the system work as expected and to confirm that the system meets all relevant standards, guidelines and customer requirements. The process of writing a test case can also help detect errors or defects in the system. Test cases are typically written by members of the quality assurance team (QA) or testing team, and can be used as step-by-step instructions for each system test.

Defect report: This is a document that identifies and describes a defect detected by a tester or user [9]. The purpose of the defect report is to state the problem as clearly as possible so that the developers can easily reproduce the defect and correct it. Writing software bug reports is an important skill for software testers, quality control. The defect report contains a sequence of actions that leads to an unexpected result, or the user receives an unexpected error: the system hangs, or the calculated data is inaccurate, and so on.

Troubleshooting**:** Troubleshooting is the process of diagnosing the source of a problem. It is used to troubleshoot hardware, software, and many other products. The basic theory of troubleshooting is that you start with the most general (and often the most obvious) possible problems and then narrow them down to more specific issues [10].

Few testing methodologies:

Performance testing - testing, which is carried out in order to determine how quickly a computing system or part of it works under a certain load. It can also be used to check and validate other attributes of system quality such as scalability, reliability, and resource consumption. This method will be described in more detail in Section 6, using the TCP protocol stack as an example.

Security testing: Application security for IoT devices is probably the biggest problem for most users. People can use applications for IoT devices for a variety of reasons, including monitory transactions, sharing personal information, buying and selling, and more. That's why the Internet of Things app needs to have special access control tools for users that restrict access to outsiders. Without security testing, it is impossible to detect major vulnerabilities in the application of the Internet of Things device and make sure that it does not transmit confidential information to attackers.

Required hardware and software: To test the TCP protocol stack, you need to have two computers with Linux installed, as well as a number of programs: wireshark (analog of the tcpdump traffic interceptor with a convenient and clear interface, iperf – TCP / UDP traffic generator, which allows you to check system stability, dhcp – server, and also some built-in Linux utilities).

4.2 Example of test documentation

Shown as an example one of the most popular types of test documentation, namely a defect report. A defect report is needed so that developers can quickly fix errors. Defect reports should include detailed information about the platform, environment, or other technical information to create detailed description. Your defect report should be clear and easy to read. The purpose of the defect report is to resolve the issue as soon as possible so that customers can continue to use the software. Here are the main sub-items for the report:

- Summary
- Description
- Build / platform
- Playback steps
- Actual results
- Expected results

This way, the developer or manager can quickly get a clear idea of the error. The developer will be able to quickly reproduce the error using step-by-step instructions, and then correct it.

Here is an example of a report defect that describes a failed dhcp server startup and suggests a solution to this problem.

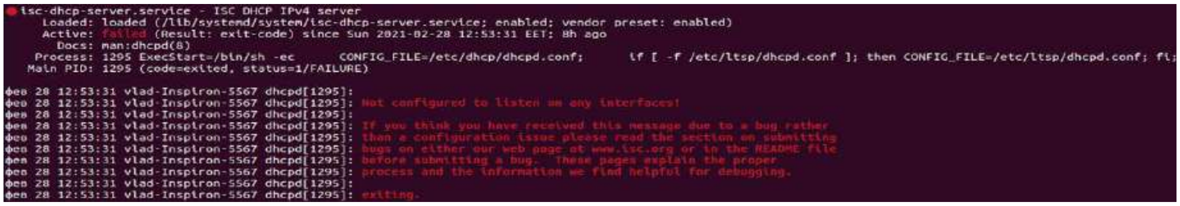
Summary: DHCP doesn’t start
OS: Ubuntu 20.04 Severity: Minor

Priority: Medium
Status: Assigned

Description: DHCP doesn’t start because dhcpd.conf isn’t configured correctly

Steps to Reproduce:

1. Configure: dhcpd.conf
2. Start dhcp: sudo dhcpd
3. Check DHCP status sudo systemctl status isc-dhcp-server

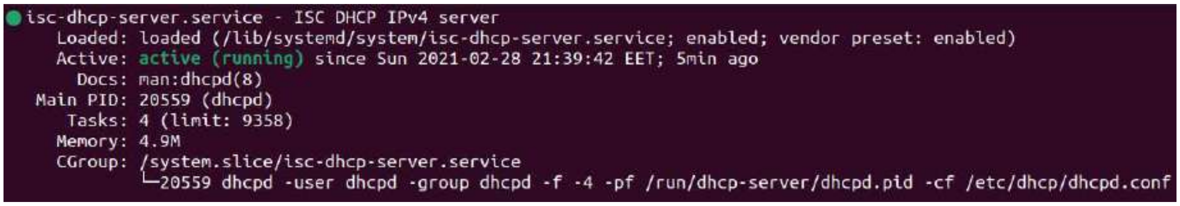


```
● isc-dhcp-server.service - ISC DHCP IPv4 server
     Loaded: loaded (/lib/systemd/system/isc-dhcp-server.service; enabled; vendor preset: enabled)
     Active: failed (Result: exit-code) since Sun 2021-02-28 12:53:31 EET; 8h ago
       Docs: man:dhcpd(8)
    Process: 1295 ExecStart=/bin/sh -ec      CONFIG_FILE=/etc/dhcp/dhcpd.conf;      if [ -f /etc/ltsp/dhcpd.conf ]; then CONFIG_FILE=/etc/ltsp/dhcpd.conf; fi;
   Main PID: 1295 (code=exited, status=1/FAILURE)

фев 28 12:53:31 vlad-Inspiron-5567 dhcpd[1295]:
фев 28 12:53:31 vlad-Inspiron-5567 dhcpd[1295]: Not configured to listen on any interfaces!
фев 28 12:53:31 vlad-Inspiron-5567 dhcpd[1295]:
фев 28 12:53:31 vlad-Inspiron-5567 dhcpd[1295]: If you think you have received this message due to a bug rather
фев 28 12:53:31 vlad-Inspiron-5567 dhcpd[1295]: than a configuration issue please read the section on submitting
фев 28 12:53:31 vlad-Inspiron-5567 dhcpd[1295]: bugs on either our web page at www.isc.org or in the README file
фев 28 12:53:31 vlad-Inspiron-5567 dhcpd[1295]: before submitting a bug.  These pages explain the proper
фев 28 12:53:31 vlad-Inspiron-5567 dhcpd[1295]: process and the information we find helpful for debugging.
фев 28 12:53:31 vlad-Inspiron-5567 dhcpd[1295]:
фев 28 12:53:31 vlad-Inspiron-5567 dhcpd[1295]: exiting.
```

Fig. 1. Example of a failed DHCP server startup

Actual result: DHCP server failed to start due to incorrect *dhcpd.conf* configuration
Expected result: DHCP server started successfully



```
● isc-dhcp-server.service - ISC DHCP IPv4 server
     Loaded: loaded (/lib/systemd/system/isc-dhcp-server.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2021-02-28 21:39:42 EET; 5min ago
       Docs: man:dhcpd(8)
   Main PID: 20559 (dhcpd)
      Tasks: 4 (limit: 9358)
     Memory: 4.9M
     CGroup: /system.slice/isc-dhcp-server.service
             └─20559 dhcpd -user dhcpd -group dhcpd -f -4 -pf /run/dhcp-server/dhcpd.pid -cf /etc/dhcp/dhcpd.conf
```

Fig. 2. An example of a successful start of a DHCP server

HowTo Fix: Configure *dhcpd.conf* right. In this case, write the subnet mask correctly

5. Results. Performance test

Performance testing is the practice of assessing how a system works in terms of response speed and stability under a given workload. Performance tests are usually performed to check the speed, stability, reliability and size of the program. The process includes such indicators of "effectiveness" as:

- Browser, page and network response time
- Server request processing time
- Simultaneous custom volumes are allowed
- CPU memory consumption; the number and type of errors that may occur in the application

In our case, testing the stability of the system will be demonstrated using bash scripts that are very easy to write and use. An example of such a scenario is shown below.:

```
#!/bin/bash
        print_usage() {
        echo "Usage: $0 ip_addr device"
                       }
        if [ $# -ne 2 ]; then
        echo "Error: Too few arguments" >&2
        print_usage
        exit 1
        fi
        iperf -c $1 -i 1 -t 30 | tee tcp_$2.txt
```

This script uses an *iperf* traffic generator that requires a client and a server. This script accepts the IP address of the server and the client. You can also select the desired script execution time and the interval at which the result will be displayed. At the output we get a text file with a traffic log.

Next, you can use the *gnuplot* script, which can be used to draw a graph that will use *.txt traffic derived from the *bash* script. Further is an example of a *gnuplot* script:

```
set xlabel "Time, sec"
        set ylabel "Mbit/sec"
        files = system("ls -1 *.txt")
        phone(f) = substr(f, 5, strlen(f) - 4)
        plot                          \
        for [file in files] file      \
        using ($3>8 ? $7 : $8)        \
        every ::6                     \
        title phone(file)             \
        with lines
        pause mouse close
```

You must make both scripts executable. To do this, use the command:

```
sudo chmod + x <file name>.
```

You can now run these scripts sequentially and get the result as a graph (fig. 1).

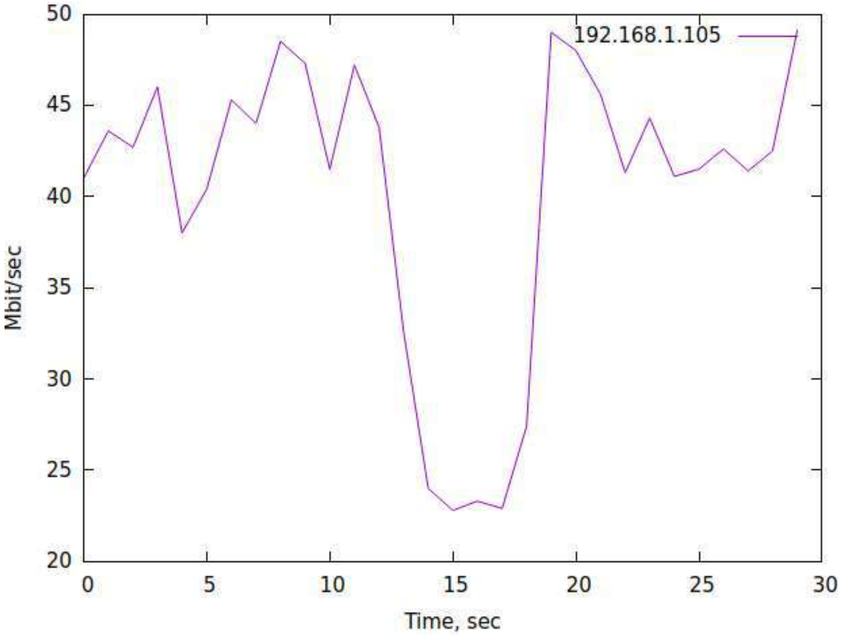


Fig. 3. Generated traffic graph

6. Conclusion

Today, the Internet of Things is one of the most popular industries. Therefore, in the development of IoT devices there is a need for detailed testing of these devices. If you do not fully test the device before selling it, many malfunctions may occur in the future, and this will result in the loss of a large amount of money for the company. Also this article discusses the basics of testing: writing test documentation, and some testing methods. The writing of the defect report on an example of start of the dhcp server was considered in more detail, and also the performance test which uses the usual bash script as a basis was demonstrated.

References

[1] “Real Life Examples of Embedded Systems,” *The Engineering Projects*, Nov. 12, 2016. https://www.theengineeringprojects.com/2016/11/examples-of-embedded-systems.html

[2] “What is embedded system?” *IoT Agenda*, 2019. https://internetofthingsagenda.techtarget.com/definition/embedded-system

[3] “How Embedded Systems Impact Your Everyday Life,” *Electronics Maker*, Apr. 13, 2018. https://electronicsmaker.com/how-embedded-systems-impact-your-everyday-life

[4] W. Wolf, B. Ozer, and T. Lv, “Smart cameras as embedded systems,” *Computer*, vol. 35, no. 9, pp. 48–53, Sep. 2002, doi: https://doi.org/10.1109/mc.2002.1033027.

[5] TechSci Research, https://www.techsciresearch.com, “Embedded Systems- The Heart of Automotive Market,” *Techsciresearch.com*, 2017. https://www.techsciresearch.com/blog/embedded-systems-the-heart-of-automotive-market/44.html

[6] “Importance of IoT Testing, what is Internet of Things testing types and process,” *PFLB*, Jun. 05, 2020. https://pflb.us/blog/iot-testing-importance/

[7] R. L. Mitchell, “The Internet of Things at home: Why we should pay attention,” *Computerworld*, Jun. 30, 2014. https://www.computerworld.com/article/2696046/the-internet-of-things-at-home--why-we-should-pay-attention.html

[8] T. Hamilton, “How to Write Test Cases: Sample Template with Examples,” *Guru99.com*, Mar. 23, 2019. https://www.guru99.com/test-case.html

[9] A. Reichert, “How to write a software defect report,” *TechBeacon*. https://techbeacon.com/app-dev-testing/write-software-defect-reports-get-results-boost-credibility

[10] “Computer Basics: Basic Troubleshooting Techniques,” *GCFGlobal.org*, 2019. https://edu.gcfglobal.org/en/computerbasics/basic-troubleshooting-techniques/1/

UDC 004.056.53
DOI: https://doi.org/10.20535/2708-4930.2.2021.244178

O. MARKOVSKYI,
O. RUSANOVA
AL-MRAYT GHASSAN ABDEL JALIL HALIL
O. KOT

ONE APPROACH TO ACCELERATE THE EXPONENTIATION ON GALOIS FIELDS FOR DATA PROTECTION CRYPTOGRAPHIC SYSTEMS

The new approach to accelerate the computational implementation of the basic for a wide range of cryptographic data protection mechanisms operation of exponentiation on Galois Fields have been proposed. The approach is based on the use of a specific property of a polynomial square and the Montgomery reduction. A new method of squaring reduces the amount of computation by 25% compared to the known ones. Based on the developed method, the exponentiation on Galois Fields procedure has been modified, which allows to reduce the amount of calculations by 20%.

Keywords: multiplication operation on Galois fields, cryptographic algorithms based on Galois Fields algebra, Galois Fields exponentiation, Montgomery reduction.

1. Introduction

The dynamic development of the Internet and computer technology has led to the emergence and widespread use of cloud technologies. These technologies provide a wide range of users with access to virtually unlimited computing power, large amounts of memory and modern software. Thus, cloud technologies can significantly increase the capabilities of the widest range of users to solve their scientific and applied problems.

On the other hand, access to these technologies has become more accessible not only to ordinary users, but also to villains, who were among the first to join the opportunities offered by cloud technology [1]. The tasks of selecting keys to existing cryptographic mechanisms for information security are well parallelized and, accordingly, effectively solved on powerful multiprocessor remote computer systems [2]. Thus, the advent of cloud technology has objectively upset the balance between the level of cryptocurrency and the resources available to villains [2]. To counteract this, there is a need to find new ways to increase the level of cryptocurrency, first of all, public key information protection algorithms. Most of these algorithms are based on the mathematical operation of modular exposition, which is performed on large numbers (2048 or 4096).

One possible solution to this problem is to increase the bit size of the keys used in the corresponding cryptographic algorithms. However, such a decision will result in a significant slowdown in the computational implementation of cryptographic protection mechanisms. In particular, doubling the bit size slows down the execution of algorithms eight times [3].

Another solution to the problem of accelerating the computational implementation of cryptographic algorithms with a public key is to move to another algebraic basis, in particular to the algebra of Galois fields [4]. Operations in these fields are performed an order of magnitude faster due to the lack of hyphens. Further acceleration can be achieved through the use of additional resources. To use them, it is necessary to develop new methods aimed at accelerating the execution of the exposure operation in the Galois fields.

Thus, the scientific task of accelerating the execution of the exposure operation in the Galois fields is relevant at the present stage of development of information technology.

2. Problem statement and review of methods for its solution

The tendency to expand the use of exposure to Galois fields in modern mechanisms of cryptographic protection of information stimulates intensive research aimed at accelerating the performance of multiplicative operations on numbers whose bit size far exceeds the bit size of the processor [5].

In the transition in cryptographic using to the algebra of Galois fields from traditional algebra, in order to distinguish operations in each of these algebras use different notation. In particular, in the algebra of Galois fields, the traditional addition is replaced by the addition operation in the Galois fields, which is denoted by the symbol '$\oplus$' is a bitwise operation "Excluding OR" (XOR). In the algebra of Galois fields there is no subtraction operation usual in traditional algebra. The basic multiplication operation in the Galois field algebra consists of two operations: polynomial multiplication and reduction using a base polynom of the field [6]. The polynomial multiplication is denoted by the symbol '$\otimes$', and the reduction operation consists in calculating the remainder of the polynomial division of the result of the polynomial multiplication by the Galois field forming the polynomial $P(x)$ [6]. The operation of calculating the remainder of the polynomial division of the number A by P is denoted as A rem P. The product of the numbers A and B in the Galois fields is denoted as $A\otimes B$ rem P. The operation of calculating exponents on Galois fields, is the calculation of the polynomial remainder from the division of the product E of the numbers A by the polynomial field P is denoted as $A|^E$ rem P in contrast to the modular exposition A^E mod M adopted in traditional algebra [6].

The existing methods of exposition on Galois fields are based on two classical algorithms: from the lower and upper bits of the code the exponents $E = \{e_n, e_{n\text{-}1}, \ldots, e_0\}$ for any $j \in \{1, 2, \ldots, n\text{-}1\}$; $e_i \in \{0, 1\}$. In both mentioned algorithms it is impossible to perform several cycles simultaneously [7]. The advantage of the lower-bit exponential algorithm is the ability to partially parallelize calculations within a single cycle. This allows you to increase the speed of the algorithm by 1.5 times.

Further acceleration of the exponent operation in Galois fields is carried out by reducing the execution time of multiplication in the field [8]. This operation consists of polynomial multiplication and reduction. The operation of polynomial multiplication of n-bit numbers requires $0.5n$ logical addition operations and n shift operations to calculate the product. Taking into account that the execution time of the logical addition command is approximately the same as the execution time of the shift command, we can assume that the implementation of polynomial multiplication is determined by the execution time of $1.5n$ logical operations [9]. As the main reserve for accelerating multiplication in Galois fields, most researchers consider the reduction operation [10].

The polynomial reduction operation is performed by adding a number corresponding to the forming polynomial to the current remainder. This operation includes determining the position of the highest digit of the current remainder, shifting the code forming the polynomial, logically adding it to the current remainder [11]. Thus, to perform the reduction, you need to perform an average of n bit testing operations, $2n$ offset operations (offset code of the forming polynomial and test code containing one unit), as well as $0.5n$ logical addition operations. The total average number of logical operations to perform reduction by dividing polynomials is $3.5n$.

Further increase in speed is achieved by accelerating the reduction. Most of the known methods are based on the use of precalculations dependent on the unchanging polynomial P [12], which in cryptographic information protection systems is part of the public key and, accordingly, rarely changes.

In acceleration methods based on the use of this property of the forming polynomial, the residues from the division of the codes $2^{n+1},\ldots,2^{2\cdot n}$ by the forming polynomial $P(x)$: $T_1 = 2^{n+1}$ rem P, $T_2 = 2^{n+2}$ rem $P,\ldots,T_{\mathrm{n}}= 2^{2\cdot n}$ rem P are preliminarily calculated. The calculated codes are stored in the table memory of precalculations. The reduction is reduced to the addition of tabular codes, which correspond to the units in the highest n digits of the code of the polynomial product. Thus, due to the use of precalculations, it is possible to reduce the average number of logical operations to implement the reduction to $1.5n$. The total average number of logical operations for multiplication on Galois fields is $3n$.

Another way to accelerate the reduction in Galois fields is proposed in [13, 14] and its essence is to adapt the Montgomery technology known in traditional algebra to the features of the algebra of

Galois fields. Using Montgomery technology, the average number of logical operations for the computational implementation of multiplication in Galois fields was reduced to $2n$ rounds [15].

An analysis of both classical Galois finite-field exposition algorithms shows that 2/3 of the computational volume is accounted for by the square operation. Therefore, the most promising way to accelerate these important cryptographic calculations is to conduct research aimed at reducing the computational complexity of squaring in the Galois fields.

3. Purpose and objectives of research

The aim of the study is to accelerate the calculation of the exponent on the finite Galois fields in software and hardware implementation by reducing the number of logical operations required for squaring in the Galois fields.

To achieve this goal, the study solves the following tasks:

– analysis of the features of symmetry of operations when squaring in Galois fields and finding ways to use them to accelerate squaring – the basic operation of exposition in Galois fields;

– development of a method of accelerated squaring in the Galois fields, the difference of which is to eliminate duplication of calculations, thereby reducing the computational complexity;

– development of a modified exposition procedure in Galois fields using the accelerated method of squaring;

– evaluation of the effectiveness of the proposed method of squaring in the Galois fields and exposition in terms of accelerating their computational implementation.

4. The method of accelerated elevation to the square in the Galois fields using the Montgomery reduction

Twothirds of the computational volume that makes up the Galois field exponentiation, as well as the traditional modular exposition, is the squaring operation [8]. Therefore, it is important to find opportunities to accelerate this dominant component of the implementation of cryptographic mechanisms in the transition from traditional modular exposure to perform this operation in Galois fields.

The property of a polynomial square and the application of the Montgomery reduction can be considered as the main reserves for the acceleration of calculations related to the squaring in the Galois fields.

The property of a polynomial square is that the addition to the square of the number $A = a_{n-1}\cdot 2^{n-1} + a_{n-2}\, 2^{n-2} + \ldots + a_1 2 + a_0$, where $\forall\ i \in \{0,1,\ldots,n-1\}$: $a_i \in \{0,1\}$ is reduced to the insertion of "zeros" between the binary digits a_0, $a_1,\ldots,a_{n-1}$ of the number A: $A \otimes A = A^2 = a_{n-1}\cdot 2^{2\cdot(n-1)} + a_{n-2}\cdot 2^{2\cdot(n-2)} + \ldots + a_1\cdot 4 + a_0$ [11]. For example, if $A = 9_{10} = 1001_2$, then its polynomial representation has the form: $A(x) = x^3 + 1$. Accordingly, the polynomial square of this number can be represented as: $A(x) \otimes A(x) = (x^3+1)\cdot(x^3+1) = x^6+x^3+x^3+1 = x^6+1$. Inserting "zeros" between binary digits gives a similar result: $A^2 = 1\ 0\ 0\ 0\ 0\ 0\ 1_2 = 65$.

Thus, the first component of squaring in the Galois fields – polynomial multiplication does not require for its implementation any operations other than shifts. Montgomery technology adapted to Galois field algebra can be used to accelerate the computational implementation of the second component, the reduction of a polynomial square [12].

To realize the above possibilities, the following method of squaring in Galois fields is proposed.

There is a number A such that $A = a_{n-1}\cdot 2^{n-1} + a_{n-2}\cdot 2^{n-2} + \ldots + a_1\cdot 2 + a_0$, and $\forall i \in \{0,1, \ldots, n-1\}$: $a_i \in \{0,1\}$. It is necessary to perform the operation of squaring this number to the square, ie to calculate $A \otimes\ A^2$ rem P, where P is the number that corresponds to the forming polynomial of the Galois field: $P = p_n\cdot 2^n + p_{n-1}\cdot 2^{n-1} + p_{n-2}\cdot 2^{n-2} + \ldots + p_1\cdot 2 + p_0$; $\forall j \in \{0,1,\ldots,n\}$: $p_j \in \{0,1\}$. Montgomery's technology involves the use of an auxiliary polynomial R, $R = 2^n$ for which the multiplicative inversion R^{-1} is determined so that $R\cdot R^{-1}$ rem P=1.

The proposed method involves the following sequence of actions:

1. The counter j cycles is set to zero: $j = 0$.

2. The number B is formed: B: $B = b_{2n-1}\cdot 2^{2n-1} + b_{2n-2}\cdot 2^{2n-2} +\ldots+ b_1\cdot 2 + b_0$ and $\forall k\in\{0,1, \ldots, 2n-1\}$: $b_k = a_{k/2}$, if k mod 2 = 0 and $b_k = 0$ if k mod 2 = 1.
3. If $b_0 = 0$, then proceed to claim 5.
4. To the current value of the code B is added modulo 2 the value of the code P, which corresponds to the forming polynomial of the code: $B = B\oplus P$.
5. A shift to the right by one bit of the value of the code B of the current result: $B >> = 1$;
6. The unit is added to the cycle counter: $j = j + 1$. If $j <n$, then there is a return to re-execution of claim 3.

Next will be show that as a result of the proposed procedure it obtained the value of the result B, which is equal to $A\otimes A\otimes R^{-1}$ rem P. If we denote by D the polynomial square $D = A\otimes A$, which is obtained by inserting "zeros" between each pair of binary digits of the number A, the value of D is equal to the initial value of B, which is formed in paragraph 2 of the developed procedure. In the process of its implementation to the value of D are added h values of the number P, where $0 < h \leq n$. The addition of the numbers P is carried out in such a way that their logical sum with the code D has zeros in n lower digits. That is, the code B', which is obtained as a result of the above procedure, excluding offsets, can be represented as: $B' = A\otimes A \oplus S$, where S is the logical sum h of shifted codes P. Offset of the obtained result B′ by n positions to the right, provided that the lower n bits of B' are equal to zero, equivalent to the multiplication of B' by the multiplicative inversion R–1 of the code R = 2n, the multiplication by which is identical to the shift operation to the left by n bits. Thus, the code B obtained as a result of the procedure described above is a reduction of the product: $B'\otimes R^{-1} = (A\otimes A\oplus S)\otimes R^{-1} =(A\otimes A)\otimes R^{-1}\oplus S\otimes R^{-1)}$. In other words, $B = B'\otimes R^{-1}$ rem $P = (A\otimes A)\otimes R^{-1}$ rem $P\oplus S\otimes R^{-1}$ rem P. Due to the fact that the second component of the sum includes as a component of the product the sum of codes P, then the value of its remainder from the polynomial division by P is zero. This means that the obtained, as a result of the proposed and described above procedure is equal to $B = (A\otimes A)\otimes R^{-1}$ rem P, which had to be proved.

The proposed algorithm can be illustrated by the following numerical example.

Let $n = 4$, $A = 11_{10} = 1011_2$, forming polynomial $P = 19_{10} = 10011_2$, an auxiliary polynomial $R = 2n = 16$, and its multiplicative inversion is equal to $R^{-1} = 14_{10} = 1110_2$. Indeed, $R\cdot R$–1 rem $P = 16\cdot 14$ rem 19 = 1. It is necessary to raise to the square of the number A on the Galois field with the forming polynomial P (x) = x4 + x + 1, which is related to the number P = 19: $A\otimes A$ rem $P = 11\otimes 11$ rem 19 = 9.

Before performing the calculations, according to claim 1 of the above procedure, the counter j cycles is set to zero, and the initial value of the number B is formed from a given number A by inserting zeros: B = 1000101.

The dynamics of transformations of variable B in the process of performing cycles of the above procedure is presented in table 1.

Table 1

Dynamics of transformations of variable B in the process of performing cycles of the proposed procedure of accelerated squaring in the Galois fields

j	The value of the variable B			
	At the beginning of the cycle	b_0	After performing step 4	After performing step 5
0	1000101	1	1000101 ⊕ 10011 1010110	101011
1	101011	1	101011 ⊕ 10011 111000	11100
2	11100	0	–	1110
3	1110	0	–	111

The number B obtained as a result of the proposed procedure of accelerated ascent to the square is equal to $A \otimes A \otimes R^{-1}$ rem P = 11⊗11⊗14 rem 19 = 7.

The true value of U of the square A on the Galois field with the forming polynomial P(x)=x^4+x+1, which corresponds to the number P = 19 can be obtained by multiplying B by R=2^n =2^4 = 16: U = 7⊗16 rem 19 = 9.

According to conducted research, the average multiplication time in Galois fields depends on the number of logical addition and shift operations.

In a known variant of multiplication in the fields of the Montgomery Reduction [14], the shift is performed on each of n cycles, so is performed n times. Operations of logical adding the multiplicand to current result are carried out when the current bit of the multiplier is equal to one. On large length, the probability that the current bit will be a one or zero, is equal to 50%, thus, the multiplicand logical addition will occur only in half of all cycles, that is $0.5 \cdot n$ times. Operations of logical adding of Galois fields base polynomial depends on low bit of current result. Thus, in average this operation is executed also $0.5 \cdot n$ times [16]. Then, the average total number of logical addition operators consist of n. Logical addition and shift operations require approximately the same time to execute. Therefore, it is advisable to calculate all operations together when calculating the time of the algorithm together. And the total number of all operations will be $2 \cdot n$.

In the developed method, the rise time to the square depends on number of shifts and number of logical addition operations. Adding a multiplicand in proposed method was replaced by the previous and disposable insertion of zeros. Shifts occur on each cycle, that is their number n. As in a well–known method, the logical addition operation number depends on the low bit of current result. Therefore, the average numbers of such operations is equal to $0.5n$. Acceleration occurs due to the exclusion of logical adding of multiplicand. Thus, in the algorithm according to the proposed method, the average total number of operations for square on Galois field calculation is $1.5n$.

Compared to the time of execution of a previously existing multiplication algorithm by Montgomery method [15], the number of operations decreased from $2 \cdot n$ to $1.5 \cdot n$, that is, by 25%. The conducted experimental studies were obtained by the theoretical evaluation.

5. Organization of the calculation of the exponent in the Galois field using the proposed method of squaring

As noted above, squaring in the Galois field is about 2/3 of the process of calculating the exponent in the Galois field – the basic operation of a wide class of cryptographic algorithms.

Accordingly, the squaring by the proposed method, which combines the use of a polynomial square and the Montgomery reduction, can be effectively used to accelerate the exposure in the Galois fields. The squaring operation performed by the proposed and described method is hereinafter referred to as KM (Montgomery Square) in contrast to the known multiplication scheme in Galois fields using Montgomery recursion [13] which is denoted as MM.

Modified in this way the exposition procedure on Galois fields, the calculation of A^{E} rem P involves performing precalculations before the start of cycles of sequential processing of bits of the exponent code. Montgomery technology determines the use of the auxiliary polynomial R(x) and its multiplicative inversion $R^{-1}(x)$. The number R corresponding to the polynomial $R(x) = x^n$ is defined as R = 2^{n}; accordingly, the multiplicative inversion $R^{-1}(x)$ is correlated with the number R^{-1} such that $R \otimes R^{-1}$ rem P = 1. In addition, the reduction technology according to the Montgomery method involves performing precalculations before exponentiation, namely: calculation $G = R$ rem $P = R \oplus P$ and

$D = R|^2$ rem P, as well as Z = MM (A, D) = $A \otimes D \otimes R^{-1}$ rem P. It is obvious that the values of G, D depend only on the Galois polynomial field, so they are calculated only once and can be used to expose different numbers provided that the Galois polynomial field is constant. The calculation of the number Z precedes each exposure in the Galois fields due to the fact that it depends on A.

Formally, modified as above, the Galois finite field exposure procedure using Montgomery reduction and accelerated squaring consists of the following sequence of actions:

1. The counter of h cycles is set in n: $h = n$ so that it indexes the highest unit digit of the code of the exponent E.

2. Using the developed method of accelerated ascent to the square, the value of the square G is calculated: G = KM (G).

3. If the current h-th bit of the code of the exponent E is equal to one $e_h = 1$, then the multiplication operation with Montgomery recursion obtained in the previous step of the result G on Z: G = MM (Z, G).

4. The value of the cycle counter $h = h–1$ is decremented. If $h \geq 0$, then the return is performed for re-execution of claim 2.

5. The final result is obtained by multiplying in the Galois field using the Montgomery reduction of the obtained value of G per unit: G = MM (G, 1).

The operation of the described modified Galois field exposure procedure using the accelerated squaring based on the Montgomery reduction can be illustrated by the following example. Let it be necessary to calculate A^E rem P, and $A = 12$, $E = 13$, and the Galois field-forming polynomial has the form $P(x) = x^4 + x + 1$, if corresponds to the number $P = 19$, then n = 4. Then Montgomery technology involves the use of an auxiliary polynomial $R(x) = x^4$. Its multiplicative inversion $R^{-1} = 14$. It is easy to calculate the value 12^{13} rem 19 = 8 by performing exponentiation by the classical algorithm without using Montgomery reduction.

The values $G = R \oplus P = 16 \oplus 19 = 3$ and $D = R|^2$ rem $P = 5$ are calculated in advance. Z = MM (A, D) = $12 \otimes 5 \otimes 14$ rem 19 = 7 is calculated immediately before the exposition.

After setting the counter h to the initial value 4 within stage 2 of the procedure, the square of the constant initial value $G = 3$: G = KM (3) = 3. Since the current digit e_4 of the code of the exponent E is equal to one, if $e_4 = 1$, the result under item 3 is multiplied by the Montgomery method by the value of Z: G = MM (Z, G) = MM (3,7) = 7. Then decreases by one the value of the counter $h = 3$ and, since it is not equal to one, the return for re-execution of paragraph 2.

In this paragraph 2 is the elevation to the square of the value obtained in the previous cycle G: G = KM (7) = 2. Since $e_3 = 1$, it is performed in paragraph 3, in which the result is multiplied by Z: G = MM (Z, G) = MM (2,7) = 11. Again subtract the unit from the counter h and return to claim 2.

When $h = 2$ is raised to the square of the result: G = KM (11) = 7. Since $e_2 = 0$, then paragraph 3 is skipped and decrements the value of the counter, resulting in h becomes equal to 1. Accordingly, the return to re-execution is realized item 2 within which the previously obtained result is squared: G = KM (7) = 2. Since the least significant bit of the exponent is equal to one, if $e_1 = 1$, the multiplication is performed: G = MM (Z, G) = MM (2, 7) = 11. Then decreases by one the value of h, which becomes equal to zero. This means that the exposure cycles in the Galois fields are complete. Finally, item 5 is performed – correction of the obtained result $G = 11$ by multiplying it by one: G = MM (G, 1) = MM (11,1) = 8. The obtained value corresponds to the true value 12^{13} rem 19 = 8.

The exposition operation consists of n cycles. In each cycle, the developed procedure of accelerated squaring to the square is performed, which is realized, on average, in $1.5 \cdot n$ logical operations. In addition, on average, 0.5·n multiplication operations are performed on Galois fields with Montgomery reduction, each of which, on average, requires $2 \cdot n$ logical operations to implement. In general, the average number of logical operations required for exposition on Galois fields using the proposed method is $2.5 \cdot n^2$.

Based on the fact that in the known scheme [11] of exponentiation on Galois fields with Montgomery reduction, the average number of logical operations is $3 \cdot n^2$ we can conclude that the proposed method can speed up the process of exposure to Galois fields by 20%, due to saving $0.5 \cdot n^2$ logical operations. With a typical value of $n = 2048$ for practical applications, this is 254,000 operations. Experimental studies have shown that the real acceleration of exposure in Galois fields is in the range of 18 – 22%.

6. Conclusions

As a result of research aimed at accelerating the execution of the basic for a wide range of cryptographic mechanisms of the exponentiation operation in Galois fields, a new method of accelerated squaring using Montgomery reduction is proposed.

The developed method is based on the properties of a polynomial square in Galois fields and allows to reduce by 25% the number of logical operations in comparison with the use for calculation of the multiplication square in Galois fields with Montgomery reduction. Based on the developed method, a modified exposition procedure on Galois fields with Montgomery reduction is proposed. Theoretically and experimentally it is shown that the use of a modified procedure can reduce by 20% the number of logical operations and accordingly accelerate the exposure.

The proposed solutions not only speed up the calculations, but also provide their simplification, which determines their focus primarily on the hardware implementation of cryptoprocessors.

References

[1] M. M. Boroujerdi and S. Nazem, “Cloud Computing: Changing cogitation about computing,” *World Academy of Science, Engineering and Technology*, vol. 58, pp. 1112–1116, Oct. 2009.

[2] M. Armbrust *et al.*, “A view of cloud computing,” *Commun. ACM*, vol. 53, no. 4, Art. no. 4, 2010, doi: https://doi.org/10.1145/1721654.1721672.

[3] B. Schneier, *Schneier's Cryptography Classics Library: Applied Cryptography, Secrets and Lies, and Practical Cryptography, and Malicious Cryptography.* John Wiley & Sons Inc, 2007, p. 816.

[4] O. S. Zenzin and M. F. Ivanov, *Standard of cryptographic data protection of the XXI century – AES. Finite fields.* Kudic-Obraz, 2002, p. 174.

[5] O. P. Markovskiy, Z. Leftherios, and V. R. Maksymuk, “Galois Fields Algebra Utilization for Implementation of the Conception of Zero-Knowledge Under Identification and Authentication of Remote Users,” *Èlektron. model*, vol. 6, no. 39, pp. 33–46, Dec. 2017, doi: https://doi.org/10.15407/emodel.39.06.033.

[6] M. M. Postnikov, *Foundations of Galois Theory*. Sankt-Petersburg: BXV-Peterburg Press, 2011, p. 411.

[7] E. M. Popovici and P. Fitzpatrick, “Algorithm and architecture for a Galois field multiplicative arithmetic processor,” *IEEE Transactions on Information Theory*, vol. 49, no. 12, pp. 3303–3307, doi: https://doi.org/10.1109/TIT.2003.820026.

[8] K.G. Samofalov, O.P. Markovskyi, and A.S. Sharshakov, “The method of accelerated implementation of exponentiation on Galois fields for data protection systems,” *Problems of informatization and control. NAU*, vol. 2, no. 33, pp. 143–151, 2011.

[9] O.P. Markovskyi, S. Mehmali, and G.V. Isachenko, “Technology of digital signature DSA based on Galois Fields Arithmetics,” *Herald of of National Technical University of Ukraine “KPI” Informatica, control and computer technic*, no. 55, pp. 34–41, 2012.

[10] I.A. Kalmikov, E.S. Stepanova, and K.T. Titcherov, “Development of the method of nonlinear data encryption using an exponentiation in Galois Fields,” *Modern Scientific Technologies*, no. 9, pp. 84–89, 2019.

[11] V. Osadchyy, “The Order of Edwards and Montgomery Curves,” *WSEAS TRANSACTIONS ON MATHEMATICS*, vol. 19, no. 25, pp. 253–264, May 2020, doi: https://doi.org/10.37394/23206.2020.19.25.

[12] H. Wu, M. A. Hasan, I. F. Blake, and S. Gao, “Finite field multiplier using redundant representation,” *IEEE Transactions on Computers*, vol. 51, no. 11, pp. 1306–1316, doi: https://doi.org/10.1109/TC.2002.1047755.

[13] O. S. Kot and O. P. Markovskyi, “Organization of speed up exponentiation on Galois Field using Montgomery Reduction,” *Almanac Science*, vol. 3, no. 36, pp. 34–37, 2020.

[14] O. Markovskyi, V. Masimyk, and O. Kot, “The Employment of Montgomery reduction for acceleration of exponent on Galoise fields calculation,” in *Proceeding of International Conference on Security, Fault Tolerance, Intelligence*, Kyiv, 2020, pp. 44–49.

[15] G. Hachez and J. Quisquater, “Montgomery Exponentiation with no Final Subtractions: Improved Results,” in *Cryptographic Hardware and Embedded Systems — CHES 2000*, Koç, Çetin K and C. Paar, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 293–301.

[16] S. Sherif Elfard, “Justification of Montgomery Modular Reduction,” *Advanced Computing: An International Journal*, vol. 3, no. 5, pp. 93–96, Sep. 2012, doi: https://doi.org/10.5121/acij.2012.3510.

UDC 004.383.3, 004.056, 004.8
DOI: https://doi.org/10.20535/2708-4930.2.2021.244186

V. ROMANUKE

OPTIMAL CONSTRUCTION OF THE PATTERN MATRIX FOR PROBABILISTIC NEURAL NETWORKS IN TECHNICAL DIAGNOSTICS BASED ON EXPERT ESTIMATIONS

In the field of technical diagnostics, many tasks are solved by using automated classification. For this, such classifiers like probabilistic neural networks fit best owing to their simplicity. To obtain a probabilistic neural network pattern matrix for technical diagnostics, expert estimations or measurements are commonly involved. The pattern matrix can be deduced straightforwardly by just averaging over those estimations. However, averages are not always the best way to process expert estimations. The goal is to suggest a method of optimally deducing the pattern matrix for technical diagnostics based on expert estimations. The main criterion of the optimality is maximization of the performance, in which the subcriterion of maximization of the operation speed is included. First of all, the maximal width of the pattern matrix is determined. The width does not exceed the number of experts. Then, for every state of an object, the expert estimations are clustered. The clustering can be done by using the *k*-means method or similar. The centroids of these clusters successively form the pattern matrix. The optimal number of clusters determines the probabilistic neural network optimality by its performance maximization. In general, most results of the error rate percentage of probabilistic neural networks appear to be near-exponentially decreasing as the number of clustered expert estimations is increased. Therefore, if the optimal number of clusters defines a too “wide” pattern matrix whose operation speed is intolerably slow, the performance maximization implies a tradeoff between the error rate percentage minimum and maximally tolerable slowness in the probabilistic neural network operation speed. The optimal number of clusters is found at an asymptotically minimal error rate percentage, or at an acceptable error rate percentage which corresponds to maximally tolerable slowness in operation speed. The optimality is practically referred to the simultaneous acceptability of error rate and operation speed.

Keywords: technical diagnostics, probabilistic neural network, pattern matrix, expert estimations, clustering, performance maximization.

1. Introduktion. Technical diagnostics based on expert estimations

In the field of technical diagnostics, many tasks are solved by using automated classification [1, 2]. For this, such classifiers like probabilistic neural networks (PNNs) fit best owing to their simplicity [3, 4]. Another merit is that PNNs are relatively insensitive to outliers [5]. The PNN is so simple because it is constructed easily and trained fast. Indeed, to solve a classification problem, only a pattern matrix is required. Each column in this matrix corresponds to a class (i. e., to a state of an object which is under technical diagnostics or surveillance). The elements of the column are features of the object.

To obtain a pattern matrix for technical diagnostics, expert estimations or measurements are commonly involved [1, 2, 6, 7]. Unlike other fields of diagnostics (in medicine, where a pattern matrix is obtained from images or medical tests), expert estimations are not always reliable and may contain severe biases. Similarly, measurements may be biased due to finite accuracy of tools and methodical inaccuracy. This is why every object state is estimated by at least few experts (or measurements are repeated). Subsequently, a final set of expert estimations is grouped and it can be thought of as if each expert proposes its own pattern matrix. The pattern matrix can be then deduced straightforwardly by just averaging over those expert estimations [8, 9]. However, averages are not always the best way to process expert estimations [10]. Moreover, the PNN at its input can have more than a single representative of a state (class), i. e. a few columns in a pattern matrix can correspond to the same state. Then the PNN performance may be improved by better representing the respective states.

2. Problem statement

Due to the abovementioned reasons of the uncertainty of the PNN pattern matrix deduction, the goal is to suggest a method of optimally deducing the pattern matrix for technical diagnostics based on expert estimations. The main criterion of the optimality is maximization of the PNN performance. However, the subcriterion of maximization of the operation speed should be included as well because the pattern matrix cannot be "stretched" without a limit. Indeed, too "wide" pattern matrices will operate slower. For some technical fields (e. g., where diagnostics is fulfilled frequently), the operation speed is crucial, and thus the slowness will be unacceptable.

3. A general conception of optimizing PNNs

Denote a number of object features by F, and a number of states by S. Then the smallest possible pattern for a PNN is an $F \times S$ matrix. Nevertheless, wider matrices can also be pattern. In general, an $F \times (mS)$ matrix

$$\mathbf{P}(m) = \left[p_{ij} \right]_{F \times (mS)}, \tag{1}$$

where m ∈⊠, can be a PNN pattern matrix. In matrix (1), each state is represented with m different patterns (columns), where p_{ij} is an assessment of feature i of the object at state s by

$$s = j - S \cdot \psi\left(\frac{j-1}{S} \right) \text{ for } j = \overline{1, mS} \tag{2}$$

and function $\Psi(x)$ returning the integer part of number $x \in \mathbb{R}$ [11].

If there are L experts (group measurements), then $m \leq L$. Those m different pattern matrices can be found from clustering the initial L expert matrices. Obviously, as m increases, the respective PNN operation speed may drop. So, it is necessary to determine an ultimate natural number m_{max}, at which matrix $\mathbf{P}(m_{max})$ can be used for the pattern (the slowdown in operation speed will be hard but still tolerable), but matrix $\mathbf{P}(m_{max} + 1)$ cannot be used for the pattern due to intolerable slowdown in operation speed. This can be done by plotting (tabling) a performance time curve versus m. Instead of real pattern matrices (1) for $m = 1, 2, 3, \ldots$, it is sufficient to generate random matrices of size $F \times (mS)$ and train PNNs, whereupon the PNNs are tested (on series of vectors of F numbers, whether they are random or not).

Once a maximally possible size of the pattern matrix is determined, the respective PNNs trained on pattern matrices (1) for $m \in \{\overline{1, m_{max}}\}$ are tested. Their performance is plotted (tabled) versus m. Then a number $m^* \in \{\overline{1, m_{max}}\}$ at which performance is maximal is determined.

So, a general conception of optimizing PNNs is realized via three steps as follows:

1. To determine m_{max} $(m_{max} \leq L)$.
2. To find m clusters from those L versions of pattern matrix, for each $m \in \{\overline{1, m_{max}}\}$.
3. To determine $m^* (m^* \leq m_{max})$.

Nevertheless, it is worth to additionally note that selection of m_{max} can be kind of fuzzy. Furthermore, if the performance of a PNN trained on pattern matrix $\mathbf{P}(m^*)$ is not satisfactory, number m_{max} will be probably increased. This is expected to (at least) slightly affect the operation speed, though.

4. Experimental study

To model generation of the pattern matrix, it is convenient to use normal and uniform randomizers. First of all, a pivot for each state is generated. Denote the pivot value of feature i of the object at state s by $\breve{p}_{is}$. So, let

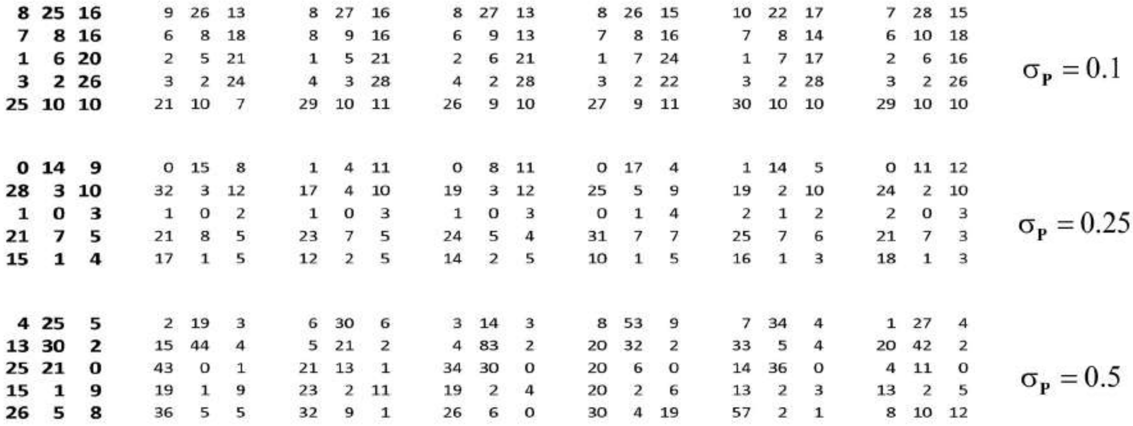
$$\breve{p}_{is} = \left|\psi\left(10\cdot\left(\xi_{is}+1.5\zeta_{is}\right)\right)\right|, \tag{3}$$

Where ξ_{is} is a random real number drawn from the standard normal distribution (with zero mean and unit variance) for feature i and state s, and ζ_{is} is a random real number drawn from the uniform distribution on interval (0; 1), $f = \overline{1, F}$ and $s = \overline{1, S}$.

At the second stage, the pivots are noised by similar randomizers. The noise is equivalent to inaccuracies of measurements and biases in expert estimations. The l-th version of estimation of feature i of the object at state s is

$$\bar{p}_{isl} = \left|\psi\left(\breve{p}_{is}\left(1+\sigma_{\mathbf{P}}\xi_{isl}\right)+1.5\zeta_{isl}\right)\right|, \tag{4}$$

where ξ_{isl} and ζ_{isl} are random numbers from the respective standard normal and uniform distributions, $l = \overline{1, L}$, and $\sigma_{\mathbf{P}}$ is a positive factor of the noise strength. Note that values (4) of expert estimations are integer because a scale for expert estimates is commonly integer or has just a few points. Three examples of generation of pattern matrix and experts' matrices are shown in Figure 1.



8 25 16	9 26 13	8 27 16	8 27 13	8 26 15	10 22 17	7 28 15	
7 8 16	6 8 18	8 9 16	6 9 13	7 8 16	7 8 14	6 10 18	
1 6 20	2 5 21	1 5 21	2 6 21	1 7 24	1 7 17	2 6 16	$\sigma_{\mathbf{P}} = 0.1$
3 2 26	3 2 24	4 3 28	4 2 28	3 2 22	3 2 28	3 2 26	
25 10 10	21 10 7	29 10 11	26 9 10	27 9 11	30 10 10	29 10 10	
0 14 9	0 15 8	1 4 11	0 8 11	0 17 4	1 14 5	0 11 12	
28 3 10	32 3 12	17 4 10	19 3 12	25 5 9	19 2 10	24 2 10	
1 0 3	1 0 2	1 0 3	1 0 3	0 1 4	2 1 2	2 0 3	$\sigma_{\mathbf{P}} = 0.25$
21 7 5	21 8 5	23 7 5	24 5 4	31 7 7	25 7 6	21 7 3	
15 1 4	17 1 5	12 2 5	14 2 5	10 1 5	16 1 3	18 1 3	
4 25 5	2 19 3	6 30 6	3 14 3	8 53 9	7 34 4	1 27 4	
13 30 2	15 44 4	5 21 2	4 83 2	20 32 2	33 5 4	20 42 2	
25 21 0	43 0 1	21 13 1	34 30 0	20 6 0	14 36 0	4 11 0	
15 1 9	19 1 9	23 2 11	19 2 4	20 2 6	13 2 3	13 2 5	$\sigma_{\mathbf{P}} = 0.5$
26 5 8	36 5 5	32 9 1	26 6 0	30 4 19	57 2 1	8 10 12	

Fig. 1. The three examples of generating a 5 × 3 pattern matrix (highlighted bold on the left) and six experts' matrices (L = 6) by increasing the noise strength factor

PNN pattern matrix (1) by (2) is determined as follows:

1. For every state s data

$$\left\{\left\{\bar{p}_{isl}\right\}_{i=1}^{F}\right\}_{l=1}^{L} \tag{5}$$

are grouped into m clusters. The clustering is done by using the k-means method [12, 13]. Consequently, m centroids of these clusters are found for every state s , $s = \overline{1, S}$:

$$\left\{\left\{c_{iks}\right\}_{i=1}^{F}\right\}_{k=1}^{m}. \tag{6}$$

2. Matrix (1) is successively formed from centroids (6):

$$p_{iz} = c_{iks} \text{ by } z = s + S\cdot(k-1) \text{ for } s = \overline{1, S}. \tag{7}$$

Once pattern matrix (1) is determined, the respective PNN is trained. Then the PNN is tested using objects whose feature *i* at state *s* is

$$q_{is} = \left|p_{is}\left(1+\sigma_{\mathbf{P}}\xi_{is}^{(1)}\right)+1.5\xi_{s}^{(2)}\right|, \tag{8}$$

Where $\xi_{is}^{(1)}$ and $\xi_{is}^{(2)}$are another random numbers from the standard normal distribution. It is worth to note that $\xi_{is}^{(2)}$ implies a normally distributed shift in state *s* of a test object. This shift is the same for all the features. Thus, model (8) of the test object differs from model (4) of the expert estimation, in which every expert has its "own" shift distributed uniformly. Besides, unlike values (4) of expert estimations, values (8) are not narrowed to a scale or set because they model real-world objects whose features are not tied to any scale.

An example of diagnosing objects with seven features by four states, where 80 expert estimations are involved, is presented in Figure 2 (the PNN testing) and Figure 3 (operation speed). Figure 2 shows that the error rate percentage decreases near-exponentially as the number of clustered expert estimations is increased. Meanwhile, Figure 3 indicates that the PNN operation speed almost linearly decreases. The value of 2.095 % is an asymptotically minimal error rate percentage, and it does not change by $m = \overline{77{,}80}$. So, $m^{*} = 77$ if the respective drop of the operation speed is tolerable (from nearly 87 to 94 seconds, which is about 8 %).

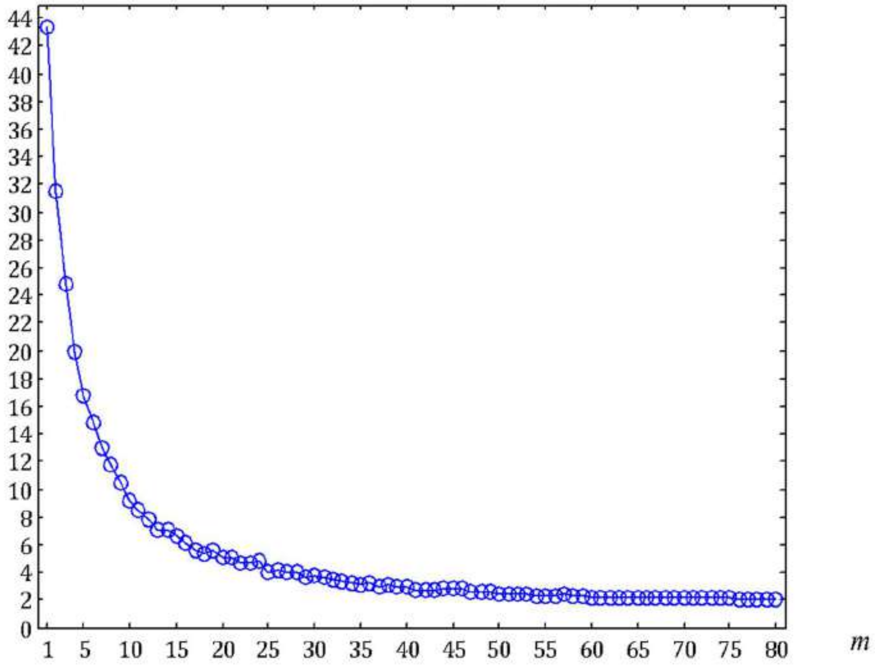


Fig. 2. The error rate percentage of PNNs by *F = 7*, *S = 4*, *L = 80* , $\sigma_{\mathbf{P}} = 0{,}25$
where every object state is tested 10000 times

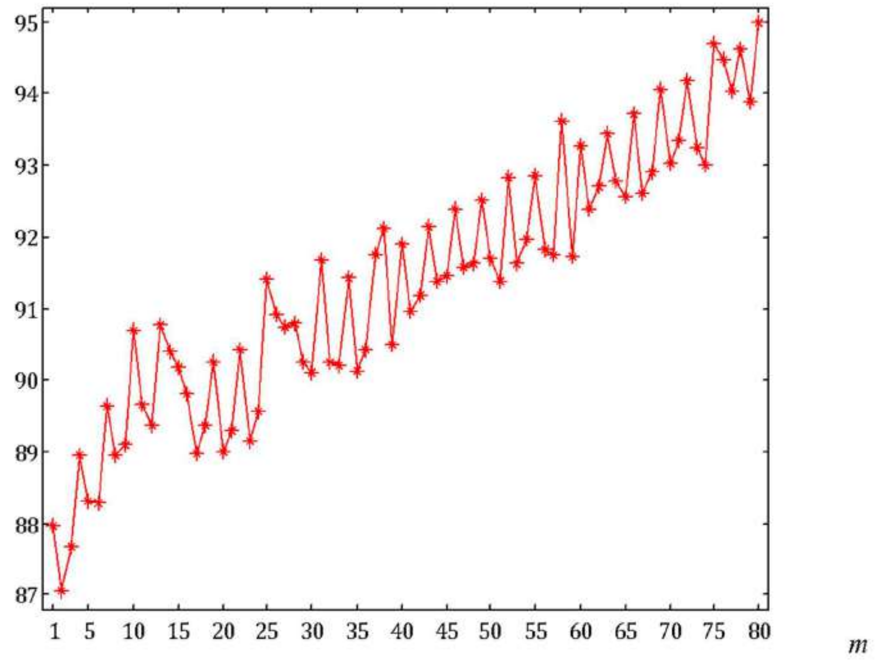


Fig. 3. Time (in seconds) spent on testing the PNNs by *F = 7*, *S = 4*, *L = 80* , $\sigma_{\mathbf{P}} = 0{,}25$
(every object state is tested 10000 times)

Another example, by smaller inaccuracies of measurements and biases in expert estimations, is presented in Figure 4 (the PNN testing) and Figure 5 (operation speed). By the same number of states, having just five features, these PNNs are far faster than those for objects with seven features. The zero error rate percentage (100 % accuracy) is achieved even by forming a pattern matrix as a concatenation of five cluster centroids (for each of the four states). So, in this particular case, $m^{*} = 5$.

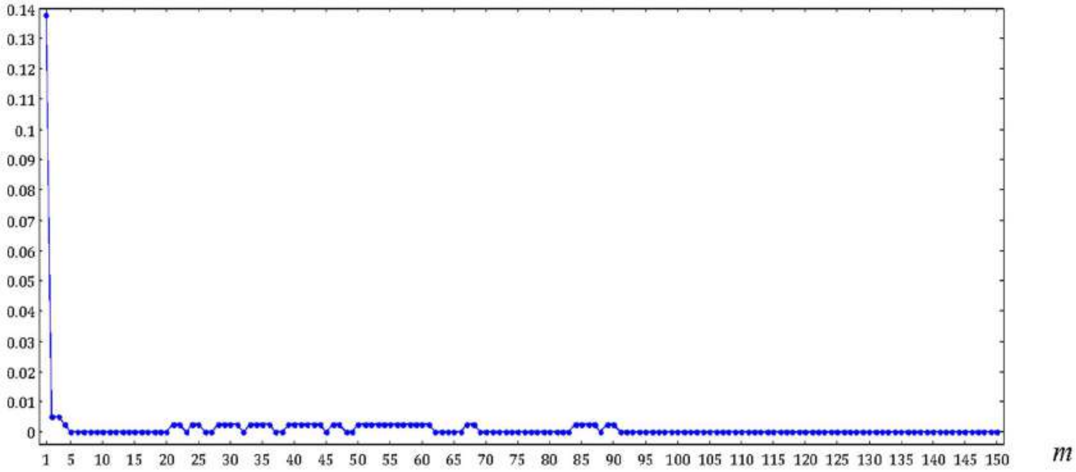


Fig. 4. The error rate percentage of PNNs by *F = 5*, *S = 4*, *L = 150* , $\sigma_{\mathbf{P}} = 0{,}1$
(smaller inaccuracies of measurements and biases in expert estimations than those for Figure 2),
where every object state is tested 10000 times

In general, most results of the error rate percentage of PNNs appear to be near-exponentially

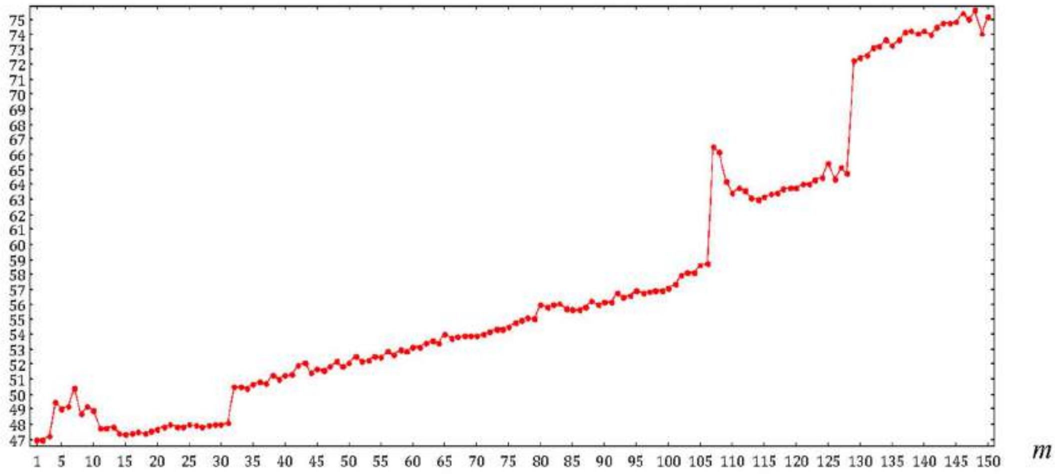


Fig. 5. Time (in seconds) spent on testing the PNNs by $F = 5$, $S = 4$, $L = 150$, $\sigma_{\mathbf{P}} = 0{,}1$
(every object state is tested 10000 times),
where computation speed artifacts are observed easier than in Figure 3

decreasing as the number of clustered expert estimations is increased. This also holds true by modeling expert estimations with varying $\sigma_{\mathbf{P}}$ and other three factors in (4) by (3), and by testing PNNs with object features (8) varying $\sigma_{\mathbf{P}}$ and the factor at $\xi_s^{(2)}$ as well. Therefore, the example in Figure 2 is a typical performance of a set of PNNs versus the number of clusters per state.

5. Discussion

While the pattern matrix is determined by (5) — (7), which is the general approach, the models of expert estimations and real-world objects are made intentionally specific. In fact, expert estimations are modeled as (4) by (3), and the PNN is tested with object features (8), where only the noise strength factor σ_P is left loose. The specification allows adjusting the models faster owing to the specified factors are close to the best making thus the models highly sensitive (susceptible to small changes in F, S, L, $\sigma_{\mathbf{P}}$ resulting in drastic changes in the error rate percentage).

The optimal number of clusters is found at an asymptotically minimal error rate percentage, or at an acceptable error rate percentage which corresponds to maximally tolerable slowness in operation speed. However, the pattern matrix cannot be limitlessly “stretched”. The optimality, therefore, is practically referred to the simultaneous acceptability of error rate and operation speed.

6. Conclusion

In technical diagnostics based on expert estimations for using them in PNNs, the pattern matrix is optimally constructed by grouping the estimations for every state into the same number of clusters. The clustering can be done by using the k-means method or similar. The optimal number of clusters determining the PNN optimality is found by the PNN performance maximization. If the optimal number of clusters defines a too “wide” pattern matrix whose operation speed is intolerably slow, the performance maximization implies a tradeoff between the error rate percentage minimum and maximally tolerable slowness in the PNN operation speed.

The suggested optimal construction of the pattern matrix for PNNs can be applied in technical diagnostics of complex objects like devices, buildings, bridges, machines, vessels (watercrafts and airplanes), etc., based on expert estimations of the object (current) state. Apart from technical and industrial systems, PNNs are nonetheless applicable in other domains (general engineering, social, ecological and economical systems, entertainment, surveillance), where the task is to control the state of objects whose number of features is up to a few tens or hundreds.

References

[1] R. Liu, B. Yang, E. Zio, and X. Chen, "Artificial intelligence for fault diagnosis of rotating machinery: A review," *Mechanical Systems and Signal Processing*, vol. 108, pp. 33–47, 2018, doi: https://doi.org/10.1016/j.ymssp.2018.02.016.

[2] H. Czichos, Ed., *Handbook of Technical Diagnostics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. doi: https://doi.org/10.1007/978-3-642-25850-3.

[3] T. Masters, "Probabilistic Neural Networks," in *Practical Neural Network Recipies in C++*, San Francisco (CA): Morgan Kaufmann, 1993, pp. 201–222. doi: https://doi.org/10.1016/B9780080514338.500173.

[4] A. Annema, *Feed-Forward Neural Networks*. NY: Springer New York, 1995. doi: https://doi.org/10.1007/978-1-4615-2337-6.

[5] V. Romanuke, G.A. Yegoshyna, and S.M. Voronoy, "TRAINING PROBABILISTIC NEURAL NETWORKS ON THE SINGLE CLASS PATTERN MATRIX AND ON CONCATENATION OF PATTERN MATRICES," *Scientific Papers of O. S. Popov Odessa National Academy of Telecommunications*, no. 2, pp. 86–97, Dec. 2019, doi: https://doi.org/10.33243/2518-7139-2019-1-2-86-97.

[6] B. K. Bose, "Expert systems and applications," in *Power Electronics and Motor Drives (Second Edition)*, Academic Press, 2021, pp. 765–788. doi: https://doi.org/10.1016/B9780128213605.000105.

[7] K. Tiwari, Y. Chong, K. Tiwari, and Y. Chong, "13Fusion of information from multiple robots: Fusion of Distributed Gaussian Process Experts (FuDGE)," in *Multirobot Exploration for Environmental Monitoring*, Academic Press, 2020, pp. 171–190. doi: https://doi.org/10.1016/B9780128176078.000289.

[8] V. V. Romanuke, "Fast Kemeny consensus by searching over standard matrices distanced to the averaged expert ranking by minimal difference," *Research Bulletin of NTUU "Kyiv Polytechnic Institute",* no. 1, pp. 58–65, 2016.

[9] Romanuke, Vadim V, "Hard and Soft Adjusting of a Parameter with Its Known Boundaries by the Value Based on the Experts' Estimations Limited to the Parameter," *Electrical, Control and Communication Engineering*, vol. 10, no. 1, pp. 23–28, 2017, doi: https://doi.org/10.1515/ecce20160003.

[10] H. Bast and I. Weber, "Don't Compare Averages," in *Experimental and Efficient Algorithms*, Nikoletseas, Sotiris E, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 67–76.

[11] V. V. Romanuke, "HEURISTIC'S JOB ORDER EFFICIENCY IN TIGHT-TARDY PROGRESSIVE IDLING-FREE 1-MACHINE PREEMPTIVE SCHEDULING OF EQUAL-LENGTH JOBS," *KPI Science News*, vol. 0, no. 2, pp. 64–73, Mar. 2020, doi: https://doi.org/10.20535/kpi-sn.2020.2.181869.

[12] C. Henning, *Handbook of Cluster Analysis*. S.L.: Crc Press, 2020.

[13] C. Bouveyron, G. Celeux, M. T. Brendan, and A. E. Raftery, *ModelBased Clustering and Classification for Data Science: With Applications in R*. Cambridge: Cambridge University Press, 2019. doi: https://doi.org/10.1017/9781108644181.

UDC 004.855.5
DOI: https://doi.org/10.20535/2708-4930.2.2021.244188

R. SHAPTALA
G. KYSELOV

VECTOR SPACE MODELS OF KYIV CITY PETITIONS

In this study, we explore and compare two ways of vector space model creation for Kyiv city petitions. Both models are built on top of word vectors based on the distributional hypothesis, namely Word2Vec and FastText. We train word vectors on the dataset of Kyiv city petitions, preprocess the documents, and apply averaging to create petition vectors. Visualizations of the vector spaces after dimensionality reduction via UMAP are demonstrated in an attempt to show their overall structure. We show that the resulting models can be used to effectively query semantically related petitions as well as search for clusters of related petitions. The advantages and disadvantages of both models are analyzed.

Keywords: vector space model, FastText, Word2Vec, petitions analysis, UMAP.

1. Introduction

By now, e-petitions have already matured and are incorporated in a lot of countries' governments. They are no longer experimental and citizens use them actively to make suggestions for public institutions. This is why the analysis of e-petitioning is vital to better understand the relationship between governmental systems and the public [1]. Automatic petition processing can help institutions immensely not only by filtering out noisy petitions, spam, and simply angry threats but also by aggregating people sentiments toward certain changes, events, or orders in an objective manner. Political implications of online petitions are well described in [2].

Unfortunately, a lot of effort is going into manual analysis of petitions which may lead to biased conclusions, is prone to errors, and inefficient. An example of the effort that went into the analysis is [3] where authors were searching for insights in the 'Save the Cretan landscape: Stop golf development at Cavo Sidero' online petition.

Ukraine is no stranger to the e-petition applications. Kyiv city – the capital and the largest city of Ukraine with a population of around 3 million people [4] – has a platform for submitting online petitions to the Kyiv City Council – petition.kievcity.gov.ua. An e-petition makes it possible for citizens to suggest actions to the Kyiv City Council. Our research is aimed at building a model of petitions posted on the above mentioned platform in order to be able to search for relevant petitions given a natural language query.

Similar research has been done by Hagen, L. et al. [5] where the authors have used We The People website data to uncover latent patterns in online petitions. They analyzed linguistic and semantic features of texts and built an LDA [6] model of the provided petitions. While very powerful, the LDA model is more of an exploratory tool that extracts main topics from petitions and lacks the contextual knowledge that models on top of the distributional hypothesis provide [7].

2. Vector space models

Vector space model is an algebraic model for encoding entities as vectors for the purpose of being able to find similarity of these entities as the degree between vectors. Every vector in such a model encapsulates the semantic structure of an object so that similar objects end up having small degrees between their vectors. The degree is also called similarity. It shows how similar are objects in the vector space instead of the distance between them. The most commonly used similarity function for vector space models is cosine similarity (1). Other noteworthy mentions are Euclidean similarity and Jaccard similarity which may describe similarity between terms better in some cases or applications, for example, in hierarchical vector space models [8]. For our experiments, for petition vector space model we used cosine similarity because it is the similarity measure that is used for underlying word vectors, which are described next. Note that identical vectors are going to have cosine similarity equal to 1.0, so the more two vectors are semantically similar, the more their cosine similarity value has to be closer to 1.0.

$$S(A,B) = \frac{AB}{||A||\,||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{1}$$

where *S(A, B)* – cosine similarity between vectors *A* and *B*; *n* – dimensionality of vectors.

To embed textual documents into vectors, models based on word embeddings, Term Frequency-Inverse Document Frequency weights, document indexing, and Latent Semantic Analysis are usually used. This paper focuses on models based on word embeddings, out of which the most popular are Google's Word2Vec [9], Stanford's Glove [10], and Facebook's FastText [11]. To better understand how to convert documents into vectors we should first step back and examine word-vector models listed previously.

The Word2Vec method takes leverage of a neural network to find word relationships from text. After the training process is finished, Word2Vec algorithm can be used to find synonyms or semantically similar words and complete sentences with missing parts. Usually, the text that is used to train such a model is huge, for example, the original paper on Word2Vec trained it on Google News Corpus with 100 billion words. Typical size of the underlying vectors is 300. Since our experiments work on a much smaller scale, we choose the dimensionality of vectors to be 100. Word2Vec has two distinct architectures, however, both of them are two-layer neural networks that make use of the distributional hypothesis. The hypothesis claims that words that occur in the same contexts tend to have similar meanings [7]. The first architecture of Word2Vec is called continuous bag-of-words (CBOW) and its goal of learning is to predict the word inside a sentence from its context – a number of neighboring words. The second architecture, called skip-gram, has the opposite learning goal – given a word inside a document predict its neighbors in a certain span. It also puts more weight on closer surrounding words than more distant ones. CBOW takes less time to train than skip-gram but models words that occur in the corpus less frequently worse.

The FastText method is an extension of Word2Vec that takes into account subword information. The authors of the algorithm model morphology by considering subword units, and representing words by a sum of its character n-grams [11, 12]. They extract all of the n-grams in the length range from 3 to 6 from words which lets the model learn prefixes and suffixes as well as other morphological information present in most of the words. This makes FastText model for word vectors expressively more powerful than Word2Vec because words that were not present in the dataset can still be embedded or queried in the vector space. On the other hand, given that FastText learns representations for subword information, for small datasets, like the one that we use, the amount of learned weights and the complexity of the model grows which may lead to less quality with limited data.

One way of training both models is to use negative sampling [13] which minimizes the log-likelihood of sampled negative instances as opposed to training on positive examples only. This method is fast, simple, and widely used for similar tasks.

This paper explores a vector space model for Kyiv city petitions, that is based on the actual texts of petitions. The main question that we addressed was the choice of an approach to create vectors of petitions given vectors of words. We first train word vectors using Word2Vec and FastText which gives us 100-dimensional embeddings for words present in petitions. For every word in the text of a particular petition, the corresponding embedding was taken and averaged across every axis to generate a 100-dimensional vector for the petition as depicted in Fig. 1.

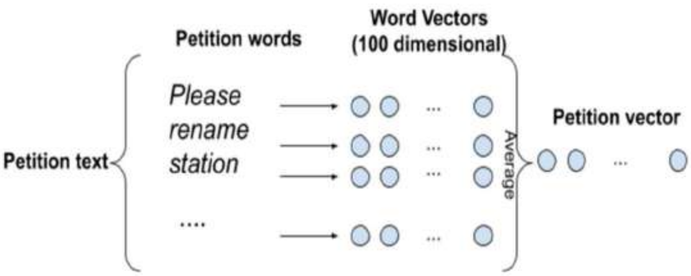


Fig.1. Petition word vectors averaging in order to get petition vector

As a result, the final vector space model is encoding the high-level meaning of the petition and can be queried for similar petitions, creating a simple exploratory data analysis tool relying purely on the semantic content of petitions. This leads to one of the drawbacks of the model: if petition description is too abstract and poorly constructed, the averaging process creates poor vector representation thus making the space less representative.

3. Word vectors

Based on the abovementioned framework, we trained two sets of word vectors: Word2Vec and FastText. The dataset that we trained on consists of 4623 petitions written in the Ukrainian language that were scraped from the petitions website. To capture semantic relationships between key entities and objects in petitions better we did several preprocessing steps:

1. stop word removal via the stop-words [14] library which provides stop words for 22 languages including Ukrainian;
2. whitespace normalization (strip any irrelevant whitespace before and after the core petition text, as well as any additional spaces in between words);
3. invisible and non-unicode characters removal;
4. infrequent tokens removal (every word that was not present in our dataset more than 20 times was removed with an intuition of being irrelevant or a spelling mistake). This made the training process more stable.

We experimented with optimal hyperparameters for both of the models, which are listed here:

1. vector dimension: 100;
2. training epochs: 1000;
3. context window: 5;
4. learning rate: 0.025.

Both models use skipgram and negative sampling as part of the internal algorithm and were trained for around 30 minutes with 4 workers on a 2.8 GHz Intel Core i5 processor. No GPU was needed because the size of the dataset is small. We used Gensim [15] as the framework for experiment implementation which allows us to build vector models with an easy-to-use, yet powerful API.

Tables 1 and 2 showcase some of the queries that became possible with the built word models.

Table 1

Top 10 closest words in Word2Vec space with their cosine distance to queried words

Queried word	**Klitschko**		**Kyiv City State Administration**	
#	***Closest word***	***Cosine distance***	***Closest word***	***Cosine distance***
1	Vitaly	0.747	Kyiv City Council	0.614
2	mayor	0.586	deputies	0.532
3	head	0.458	Kyiv City Council (abbr.)	0.518
4	head(genitive)	0.417	site	0.495
5	Kyiv(adjective)	0.374	solution	0.460
6	requirement	0.368	council	0.452
7	together	0.356	according	0.446
8	commission	0.353	District state administration	0.445
9	city	0.353	Municipal Enterprise	0.439
10	their	0.347	order	0.437

Please, note that since FastText is a modification of Word2Vec, the results of queries are similar, both capture semantic relationships, like name and job, or similar institutions of the query.

However, as mentioned before, FastText allows us to make queries for words not present in the dataset, which is a huge bonus for word vector models.

Table 2

Top 10 closest words in FastText space with their cosine distance to queried words

Queried word	**Кличко (Klitschko)**		**Kyiv City State Administration**	
#	***Closest word***	***Cosine distance***	***Closest word***	***Cosine distance***
1	Vitaly	0.724	Kyiv City Council	0.577
2	head	0.571	Kyiv City Council(abbr.)	0.567
3	mayor	0.530	deputies	0.501
4	head(genitive)	0.478	site	0.486
5	Kyiv(adjective)	0.400	order	0.468
6	commission	0.394	District state administration	0.464
7	together	0.392	council	0.446
8	Kyiv City State Administration	0.373	Municipal Enterprise	0.446
9	Dear	0.371	question	0.445
10	Vitaly(vocative)	0.365	according	0.430

4. Petition vectors

In this section, we detail the results obtained by averaging word vectors for petition contents. In order to give some qualitative measures of newly constructed petition vectors, we are going to show their visualization. The visualization is built by reducing the dimensionality of petition vectors from 100-dimensional to 3-dimensional and plotting this reduced space. The algorithm for dimensionality reduction that we use is UMAP [16] which has increased speed and better preservation of the data's global structure than other dimensionality reduction algorithms. The idea behind UMAP is to first create a high dimensional graph representation of the data and then optimize another low-dimensional graph to be as structurally similar as possible to the constructed graph. We use Tensorboard [17] to make these visualizations and explore the space manually. Tensorboard allows projecting embeddings to a lower-dimensional space via UMAP, t-SNE, or PCA. We chose UMAP because it is faster than t-SNE and more expressive than PCA [16].

For the quantitative measurement of the differences in the researched models we use the Silhouette Coefficient [18] which is defined for a single sample as:

$$S = \frac{a - b}{\max(a, b)}$$

where a – average distance between the point and all other points in the same cluster; b – average distance between the point and all other points in the nearest cluster.

The Silhouette Coefficient score is defined as the mean of Coefficients of every point. The model with better defined clusters is expected to show higher Silhouette Coefficient score. The Coefficient has a range of -1 to 1 where spaces with highly dense clusters have scores closer to 1 and with highly overlapping clusters closer to 0. Table 3 shows Silhouette Coefficient scores for Word2Vec- and FastText-based petition vectors clustered via DBSCAN [19].

Table 3

Silhouette Coefficient scores

Model	**Silhouette Coefficient score**
Word2Vec-based	0.468
FastText-based	0.004

As mentioned before, to build vector space models for Kyiv city petitions we used a simple averaging of word vectors of words present in the petition. In Fig. 2 you can see Word2Vec- and FastText-based vector spaces after dimensionality reduction visualized.

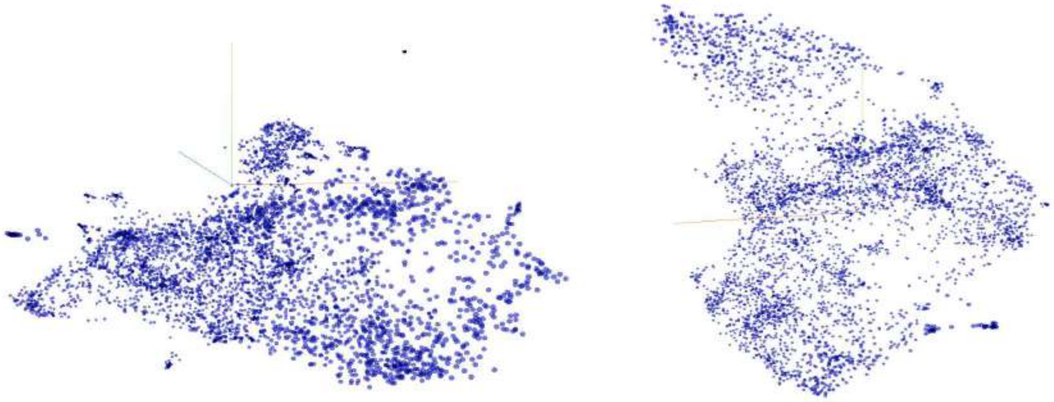


Fig. 2. Word2Vec (left) and FastText (right) petition vectors U-MAP visualizations

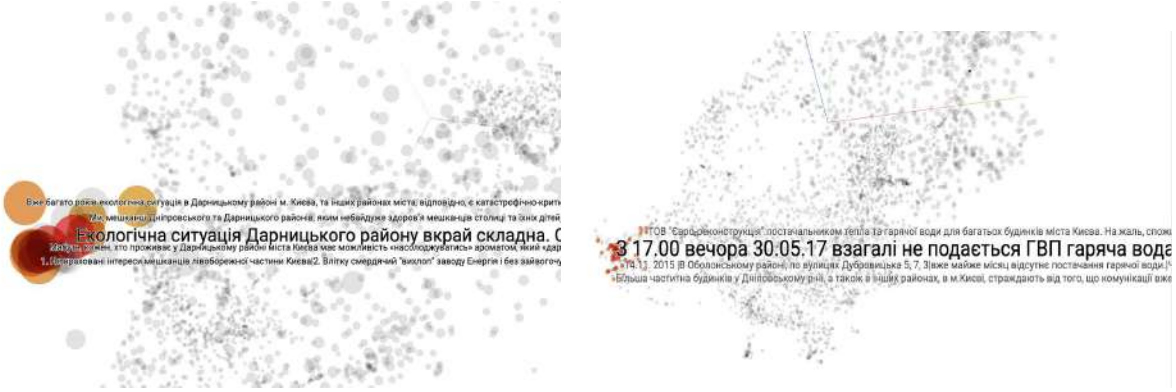


Fig. 3. Cluster of petitions about ecological situation (left) and water management (right) in Kyiv in the Word2Vec petitions space

Both spaces exhibit clustered structure and have petitions of different semantics in different parts of the space. Please, note that Word2Vec-based petition model has visibly more separated clusters than FastText-based one. This is confirmed by the Silhouette Coefficient scores listed previously, where Word2Vec based petition model had a much higher score which means the clusters that are present in its vector space are denser, while in the FastText-based model they overlap a lot.

A closer look shows that these clusters are indeed semantically divided and several well-defined groups of points exhibit similarity in the topics that they discuss. In the next section, we are going to talk in which way the two built models differ.

5. Word2Vec-based model

Upon closer inspection, Word2Vec-based model has clearly visible clusters that share some semantic meaning among them. You can see two examples of that on Fig. 3.

For more clarity on the insides of the model we provide a few queries and their similarities with the closest petitions in the dataset in the Table 4. As you can see the query that concerns about the ecological situation in Kyiv yields several petitions about poor waste processing and polluted lakes, while a query about hot water supply mostly returns complaints about it to the Kyiv city council. Overall, the quality of Word2Vec-based embedding is satisfactory for their future use for transfer learning as features to sentiment analysis classifier or any other natural language problem.

Table 4

Top 3 closest petitions in Word2Vec space with their cosine distance to queried phrases

	Ecological condition of Kyiv in Darnytskyi district		**There is no hot water supply**	
#	***Closest petition***	***Cosine distance***	***Closest petition***	***Cosine distance***
1	For many years, the ecological situation in the Darnytskyi district of Kyiv and other districts of the city, respectively, is catastrophically critical: an unbearable stench at night and in the morning as a result of poor activities of the waste processing plant "Energy"...	0.69	In the Obolonsky area, on Dubrovytska streets 5, 7, 3 there has been no hot water supply for almost a month. There is no clear answer to the question of the residents of these houses "when will there be water" from Kyivenerho and the district housing and communal services	0.841
2	it is forbidden to swim in 50 lakes and ponds, in particular Didorovsky and Mishelovsky ponds … in Darnytskyi district, … . Swimming in these lakes is not recommended due to unsatisfactory water samples. The Pleso municipal enterprise has banned beach holidays due to non-compliance with sanitary norms: according to the results of sanitary-microbiological tests of water samples. I urge you to clean Lake Sunny in Pozniaky and allow people to rest safely.	0.651	Not the first year there are complaints about the temperature of the hot water supply. The only way to add cold water to hot water arises ...	0.835
3	Apparently, everyone who lives in the Darnytskyi district of Kyiv has the opportunity to "enjoy" the aroma that "gives" us BORTNYTSKY AERATION STATION AND incinerator "ENERGY". But in addition to the unpleasant smell, companies also pose a threat to the health of EACH OF US ...	0.642	As heating and hot water services are provided by monopolists, a lever of pressure on suppliers is needed to prevent significant overstatement of tariffs. This means of influence will be the ability to install individual heating and water heating systems in any multi-storey building.	0.818

6. FastText-based model

FastText-based model shows similar semantic properties to the Word2Vec based model, their underlying core ideas are close after all. However, the overall number of visible clusters is reduced and some of them are clearly clustered on syntactic level instead of desirable semantic level. You can see this in Fig. 4, where (a) is a good semantic cluster, while (b) is similar only by having the same boilerplate start.

In Table 5 we provide two examples of FastText-based model querying. The first query is just a name of an avenue and the model could find semantically similar petitions that are mostly about its renaming process or about the process of renaming other avenues. The second example shows that the model can also find syntactically similar petitions.

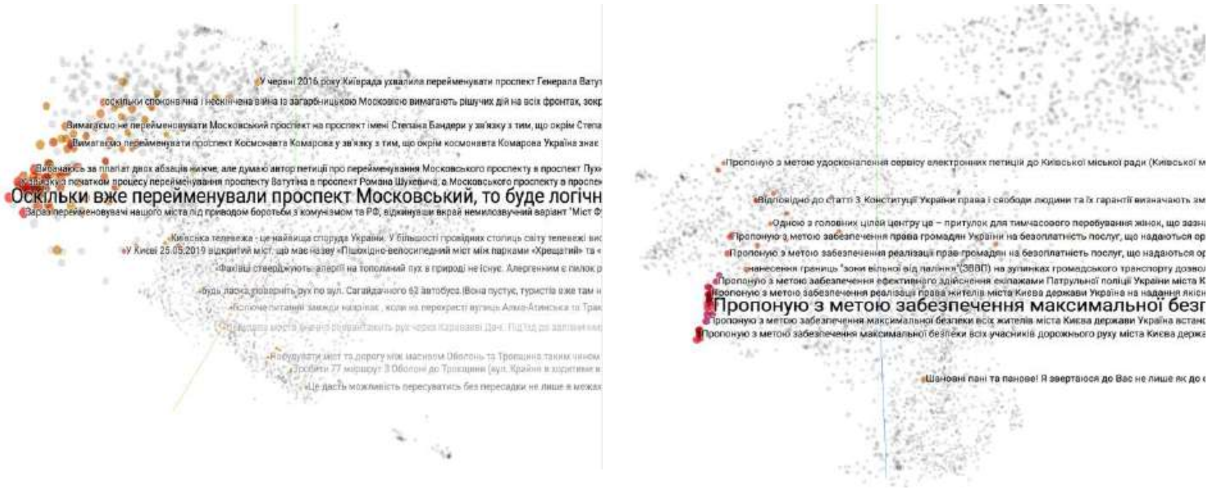


Fig. 4. Cluster of petitions about the renaming of different city objects (left) in Kyiv in the FastText petitions space and example of suboptimal separation in the FastText petitions space (right)

Table 5

Top 3 closest petitions in FastText space with their cosine distance to queried phrases

	Moscow Avenue		**In order to ensure ...**	
#	***Closest petition***	***Cosine distance***	***Closest petition***	***Cosine distance***
1	In connection with the beginning of the process of renaming Vatutin Avenue to Roman Shukhevych Avenue, and Moscow Avenue to Stepan Bandera Avenue - it would be logical to rename the Moscow Bridge, which connects the two avenues, to Troieschyna Bridge.	0.69	In order to ensure the effective implementation of the duties of the crews of the Patrol Police of Ukraine in the city of Ukraine, the state of Ukraine to provide each of them with certified devices - laser radars for measuring the speed of TruCam	0.654
2	The name of Moscow Avenue in Obolonsky and Moscow districts of Kyiv is not historical. It was given in 2003 as a friendly political gesture of the then mayor of Kyiv O. Omelchenko towards his Moscow colleague Yu. Luzhkov ...	0.651	In order to ensure the effective implementation by the crews of the Patrol Police of Ukraine of the city of Kyiv, the state of Ukraine, I propose to provide each of them with certified devices - Drager breathalyzers.	0.643
3	The name "Pravda Avenue" (Vinogradar) falls under the Law on Decommunization. In addition, when renaming it to Pavel Sheremet Avenue, it will be symbolic that the side of this avenue is Georgy Gongadze Avenue, also a deceased Kyiv journalist.	0.74	Please prohibit the movement of heavy vehicles in the city during the day, in order to reduce the destruction of asphalt pavement and to improve the organization of traffic and its safety, improve the environmental condition and increase the capacity of the road network of Kyiv	0.606

Overall, the quality of this model is less-suitable to be used in further semantically significant tasks than Word2Vec-based one.

7. Conclusion

In this paper, we proposed two methods of construction of vector space models of Kyiv city petitions, namely Word2Vec- and FastText-based word vector averaging. The main insights are that it is possible to build such vector spaces with limited data and that through our experiments Word2Vec-based algorithm was preferred since it captured more semantic representations instead of syntactic ones. This happened because, innately, FastText works on subword level and while it is useful for getting vectors of unknown character sequences, the process of word vectors averaging does eliminate this advantage, making it a liability. Quantitative results show that Word2Vec-based model is better suited for further clustering and produces denser clusters than FastText-based one.

The suggested models can be used as a stepping stone in petition analysis pipelines. The vector space models give every petition a numeric representation capturing its semantic meaning that, if included in a classification framework, can help identify citizens' attitudes toward certain events, group and deduplicate petitions with the same intent, or predict if a certain petition is going to get enough votes to pass.

Future work might include trying other aggregation functions to build petition-level vectors, like term-frequency weighting. Other possible research directions include sentiment analysis and automatic clustering of Kyiv city petitions based on built models.

References

[1] R. Lindner and U. Riehm, "Electronic Petitions and Institutional Modernization. International Parliamentary E-Petition Systems in Comparative Perspective," *JeDEM - eJournal of eDemocracy and Open Government*, vol. 1, no. 1, pp. 1–11, Sep. 2009, doi: https://doi.org/10.29379/jedem.v1i1.3.

[2] K. Böhle and U. Riehm, "E-petition systems and political participation: About institutional challenges and democratic opportunities," *First Monday*, vol. 18, no. 7, Jun. 2013, doi: https://doi.org/10.5210/fm.v18i7.4220.

[3] H. Briassoulis, "Online petitions: new tools of secondary analysis?" *Qualitative Research*, vol. 10, no. 6, pp. 715–727, Dec. 2010, doi: https://doi.org/10.1177/1468794110380530.

[4] "Population (1995-2019)," *www.kyiv.ukrstat.gov.ua*. http://www.kyiv.ukrstat.gov.ua/p.php3?c=527&lang=1 (accessed Nov. 18, 2023).

[5] L. Hagen, T. M. Harrison, Ö. Uzuner, W. May, T. Fake, and S. Katragadda, "Epetition popularity: Do linguistic and semantic factors matter?" *Government Information Quarterly*, vol. 33, no. 4, pp. 783–795, 2016, doi: https://doi.org/10.1016/j.giq.2016.07.006.

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, no. 3, pp. 993–1022, 2003.

[7] Z. S. Harris, "Distributional Structure," *WORD*, vol. 10, no. 2–3, pp. 146–162, Aug. 1954, doi: https://doi.org/10.1080/00437956.1954.11659520.

[8] M. Nickel and D. Kiela, "Poincaré embeddings for learning hierarchical representations," *Advances in neural information processing systems*, pp. 6338–6347, 2017.

[9] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, pp. 3111–3119, 2013.

[10] J. Pennington, R. Socher, and C. Manning, "Glove: Global Vectors for Word Representation," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014, doi: https://doi.org/10.3115/v1/d14-1162.

[11] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, Dec. 2017, doi: https://doi.org/10.1162/tacl_a_00051.

[12] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of Tricks for Efficient Text Classification," *CoRR*, Jul. 2016, doi: https://doi.org/10.48550/arxiv.1607.01759.

[13] T. Mikolov, K. Chen, G. s Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *Proceedings of Workshop at ICLR*, vol. 2013, Jan. 2013.

[14] A. Coenen and A. Pearce, “Understanding UMAP,” 2020. https://pair-code.github.io/understanding-umap

[15] R. Řehůřek and P. Sojka, “Software Framework for Topic Modelling with Large Corpora,” in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks.*, May 2010, pp. 45–50. doi: https://doi.org/10.13140/2.1.2393.1847.

[16] L. McInnes, J. Healy, N. Saul, and L. Großberger, “UMAP: Uniform Manifold Approximation and Projection,” *Journal of Open Source Software*, vol. 3, no. 29, p. 861, Sep. 2018, doi: https://doi.org/10.21105/joss.00861.

[17] M. Abadi *et al.*, “TensorFlow: Large-scale machine learning on heterogeneous distributed systems,” 2015. http://download.tensorflow.org/paper/whitepaper2015.pdf

[18] S. Aranganayagi, K. Thangavel, “Clustering Categorical Data Using Silhouette Coefficient as a Relocating Measure,” *Proceedings of the International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)*, December 2007, Sivakasi, India, https://doi.org/10.1109/ICCIMA.2007.328.

[19] *S. Louhichi, M. Gzara, H. Ben Abdallah*, “A density based algorithm for discovering clusters with varied density,” *Proceedings of the World Congress on Computer Applications and Information Systems (WCCAIS)*, January 2014, https://doi.org/10.1109/WCCAIS.2014.6916622.

UDC 004.383
DOI: https://doi.org/10.20535/2708-4930.2.2021.244191

P. SERHIIENKO
A. SERGIYENKO
M. ORLOVA

LOCAL FEATURE EXTRACTION IN IMAGES

The methods of the local feature point extraction are analyzed. The analysis shows that the most effective detectors are based on the brightness gradient determination. They usually use the Harris angle detector, which is complex in calculations. The algorithm complexity minimization contradicts both the detector effectiveness and to the high dynamic range of the analyzed image. As a result, the high-speed methods could not recognize the feature points in the heavy luminance conditions.

The modification of the high dynamic range (HDR) image compression algorithm based on the Retinex method is proposed. It contains an adaptive filter, which preserves the image edges. The filter is based on a set of feature detectors performing the Harris-Laplace transform which is much simpler than the Harris angle detector. A prototype of the HDR video camera is designed which provides sharp images. Its structure simplifies the design of the artificial intelligence engine, which is implemented in FPGA of medium or large size.

Keywords: FPGA, feature extraction, HDR, pattern recognition, artificial intelligence.

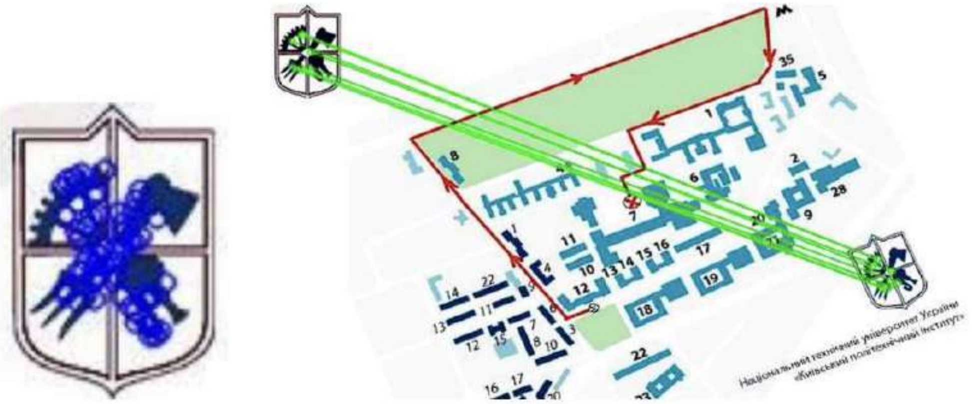
1. Introduction

The local feature points play an important role in computer vision and pattern recognition, including image matching, object recognition or clustering, image construction, object tracking, face recognition, image registration. The need for rapid detection of the feature points is manifested in applications of computer vision to obtain content-based image retrieval in robots and autonomous driving [1].

Modern approaches to identifying the local feature points of the object provide taking into account the scale, so the object can be recognized regardless of its apparent size. An object can also be characterized by a set of feature points. This allows the system to recognize the object viewed from a different angle and distance. Using the relative position of the points also provides to recognize complex objects in conditions of noise and uneven lighting.

The definition of the feature points is usually performed in two stages. At the first stage, a point is detected using the appropriate algorithm named the detector. At the second stage, the found feature point information is encoded to the descriptor.

If the image moves, rotates, scales along two axes, and the detection result remains unchanged, it means that the recognition of the points is stable, and the found points are feature points. Fig. 1 shows an example of finding a set of feature points which helps to recognize the object.


35
5
1
8
2
6
28
7
20
9
11
21
17
10
13
14
15
16
12
19
18
22
Національний технічний університет України
«Київський політехнічний інститут»

a) b)

Fig.1. Extracted feature points (a) and their use for the pattern recognition (b)

The feature point descriptors should describe the key characteristics of the image with repeatability, compactness, accuracy, and efficient representation, which are consistent and reliable in terms of scaling, rotation, display, occlusion, and lighting [2]. In this article, the methods of the local feature extraction are reviewed to select the best-fitted method for the pattern recognition in the heavy lightness conditions.

2. Image feature detection methods

Algorithms or methods for detecting and constructing a feature descriptor are divided into six categories:

– low-level detectors;
– methods based on the analysis of the brightness gradient;
– methods based on contrast analysis;
– methods based on the analysis of spatial frequency;
– learning-based methods;
– convolutional neural networks.

2.1 Low-level detectors

The low-level features are the features of the image that can be identified in it without any shape information about the object. Such features can be detected by the threshold processing, the functions sensitive to the edge, corner. The Laplace detector calculates the second-order derivative and where the result intersects the zero level, there is the edge of the form.

The Sobel detectors are based on the first derivatives of the pixel $I(x, y)$ to be considered in the coordinates (x, y). The Canny detector uses the Gaussian filtration

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y),$$

where $G(x, y, \sigma)$ is the Gauss kernel, σ is the filtering scale, and * is the convolution operator. Then, the image with the highlighted edge features is calculated as

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma). \tag{1}$$

This filter is an approximation of the Laplace filter but is simpler in calculations. This advantage is utilized in many complex detectors.

Because the derivative function is sensitive to noise, the rejection of weak results, median filter, low-pass filter, and Gaussian filter are often used for the resulting improvement. And the limit of the rejection is usually selected manually.

The comparison of edge detectors highlights their shortcomings: incomplete contours of the closed figures, the need for selective thresholds, sensitivity to noise. In addition, the threshold should be selected separately for regions with different local lighting [3].

The detector with phase congruence operates on the principle that at the edge point the sinusoidal components of the signal obtained by the Fourier transform or wavelet transform have the same phase. But this detector is also sensitive to noise and has low localization accuracy [4].

If the edge of an object is derived, then the angle and curvature detectors can find the feature points in it. The curvature is defined as the rate of change of the direction of a flat curve that describes a figure. But in all cases, finding a closed curve and its analytical description is a difficult task [5].

The degree of the curvature can be obtained by considering changes in brightness along with alternate directions in the image. This is the basic idea of the Moravec angle detection operator. This operator calculates the average change in image intensity when the analysis window is shifted in several directions. That is, for a pixel $I(x, y)$ which stays in the centrum of a window $w(x,y)$ we have a detector output, which is a measure of belonging to the line

$$E_{uv}(x,y) = \Sigma\square(I(i, j) - I(u+i, v+j))^2.$$

The shifts (u, v) are performed in four directions: (1,0), (0, −1), (0,1) and (−1,0). This equation is an approximation of the brightness autocorrelation function in the (u,v) direction. If the pixel belongs to a straight line, the value of $E_{uv}(x, y)$ is small for the offset along the line and large for the offset perpendicular to the line. The disadvantage of the detector is that it takes into account only a small set of possible directions [6].

This problem is solved in the Harris angle detector [7]. The autocorrelation of the luminance gradient along the horizontal $A(x, y)$, the vertical $B(x, y)$, and the mutual correlation $C(x, y)$ are calculated. And the measure of belonging to the line in the direction (u, v) is calculated as

$$E_{uv} (x ,y) = A(x, y)u^2 + 2C(x, y)uv + B(x, y) v^2.$$

To normalize this function, it is rotated to the angle determined by the eigenvector (α, β):

$$F_{uv} (x, y) = \alpha^2 u^2 +\beta^2 v^2.$$

According to the properties of (α, β), if the point (x, y) defines the boundary of a straight line, then one value is large and the other is small. If a point defines a boundary with high curvature, both values are large. Then, the curvature degree is defined as

$$c(x, y) = \alpha\beta - k(\alpha + \beta)^2. \quad (2)$$

This method is used in many methods described below for its effective detecting of the lines and corners. However, it has rather high complexity and needs the floating point computations.

2.2 Methods based on the analysis of the brightness gradient

The brightness gradient is measured as the intensity of the derivative in some direction. Each feature point is characterized by a set of such gradients in some of its neighborhoods. Therefore, many methods are based on deriving these gradients as orientation histograms.

The most famous and effective method is the scale-invariant feature transform (SIFT) one [8]. The method is based on detecting the characteristic points based on the concept of the scale space, ie., the relative position of the feature points should be preserved in images with different scales and rotations. The scale space is most often determined by images, which are successively smoothed by a Gaussian filter. A set of difference images is formed using the formula (1), and then, a pyramid of images is formed at different scales σ.

The SIFT method includes four stages. At the extremum detection stage, a pyramid of difference-scale images is built. In these images, the local maxima/minima as centers of characteristic points are looked for. The pixel found must be larger/smaller than the neighboring pixels in this layer and in the adjacent layers of the pyramid as is shown in Fig.2.

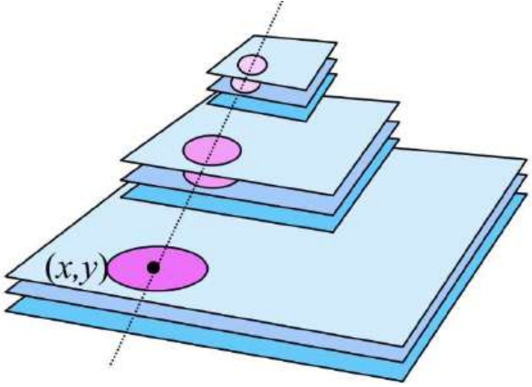


Fig. 2. The pyramid of images at different scales σ and the feature point in it

At the stage of the key point localization, the unstable feature points are filtered out, ie., the points of low contrast and those that are not angular or spots. The angles are detected by the Harris detector (2).

At the orientation search stage, the circular histogram of luminance gradients around the found point is constructed. The normalizing angle is determined, to which the local image around the center (x, y) is returned. It is possible to find 2 or 3 alternative angles, then 2 or 3 descriptors are calculated for this point.

At the last stage, the feature point descriptor is calculated and coded. A set of orientation histograms is created for the windows near the point (*x, y*) with 8 bins in each, resulting in a descriptor vector of 128 elements.

The SIFT method is the most reliable of many methods for determining the characteristic points, but also the most complex one. A large number of similar but much simpler methods have been developed, which are based on it and mentioned below.

The method of the speeded-up robust features (SURF), in contrast to SIFT, uses simple (coefficients only 0, ±1, ±2), but effective Gaussian filters that calculate edge detection (1) at different scales. But the large-scale images are not decimated. The central pixel (*x, y*) of the characteristic point is found as the maximum of the determinant of the Hess matrix. The elements of this $2 \cdot 2$ matrix are the responses of various edge filters with the Haar wavelets. Due to the simplicity of calculating the filters, a high speed of finding the feature points is provided. Although the number of detected points is less than in the SIFT method [9].

The method of the speeded-up robust features (SURF) for the formation of these features uses Haar-like filters that are easy to calculate. But the method does not use the normalization, as in SIFT, which creates the incorrect descriptors [9].

The descriptive expressive robust features (DERF) method uses a model with the spatial distribution of pixels and their processing, which simulates the properties of the ganglion cells and the retina of primates [10].

The algorithm for constructing the SIFT descriptor, which is invariant to affine transformations (ASIFT), is an improvement of the original SIFT algorithm due to its complexity. To simplify the algorithm, it combines the best parts of the SURF and SIFT algorithms [11].

In the DAISY method, eight orientation maps *Go*(*x, y*) are calculated, one for each quantized direction, where *Go*(*x, y*) is equal to the value of the positive image gradient in the pixel *I*(*x, y*) for the direction *o*. Then, the orientation maps are formed with different scales by filtering with Gaussian filters of different sizes as in the SIFT method. This improves speed and reliability, as well as insensitivity to affine transformation and brightness changes. In this case, the convolution with a large Gaussian core is simplified by building a pyramid of images. The feature point is described as a combination of normalized vectors with a large amplitude of all scales. [12].

The BOLD algorithm first finds the line segments, then generates geometric primitives as the angle between pairs of adjacent segments. The set of such primitives provide the consistency of the method in terms of rotation, mapping, scale, and noise with high informativeness [13].

2.3 Methods based on the contrast analysis

The methods based on the contrast analysis are applied to compare the intensities of pixels sampled at the different locations.

The features from the accelerated segment test (FAST) approach [14] are based on simple tests of 16 points around a pixel with *x, y* coordinates. The points are located along the circle, and if the brightness of the pixels in a particular pattern of points exceeds the brightness of *I*(*x, y*), the pixel is denoted as a feature point. The method is quite fast. But the dependence on the threshold and not taking into account the orientation reduces its reliability. Machine learning is used to improve it.

The binary robust invariant scalable keypoints (BRISK) method uses the AGAST angle detector [15], which is an accelerated version of the FAST detector. The detector is implemented as a logical decision tree with a root corresponding to the central pixel *I*(*x, y*) and branches leading to the vertices of adjacent pixels. For scale invariance, a pyramid of images with different scales obtained by the Gaussian blur is used. The symmetrical patterns with sampling points in concentric circles are used to describe the features. Then, up to 512 comparisons of pixels that are spaced from the center pixel are made. For each such comparison, the difference of the vectors leading to these pixels is calculated. These vectors are then averaged to find the dominant direction of the gradient, local gradients which represent some shape inside the analyzed window. Finally, this template is scaled and rotated. BRISK requires much more computation and more memory [16].

In the method of rapid determination of the feature points on the properties of the retina (Fast Retina Keypoint, FREAK) [17], an analogy with the work of the retina in the human visual system was used. The image is smoothed using a Gaussian filter. A window with near and far fields is created for the point under consideration. The binary descriptor is formed as a set of thresholds of differences in brightness of field pairs. Among the many options for comparing fields, the authors selected 512 of the most effective, which are used to build a descriptor. These pairs provide a highly structured pattern that mimics the jumping search of human eyes when looking at an object. The orientation of the feature point is calculated using local gradients among the selected pairs. As a result, the method decreases the computing time by two degrees of magnitude comparing to SIFT.

A similar approach, which is based on a set of binary robust independent elementary features (BRIEF), uses a brightness comparison detector and has a high speed [18].

2.4 Methods based on the analysis of the spatial frequency

At the two-dimensional transformation of space where the feature point is presented, it is possible to receive its orthogonal parameters. The Fourier descriptor is a set of two-dimensional Fourier transform coefficients of a shape region. The generalized Fourier descriptor (GFD) is obtained by pre-normalizing the image and converting it into polar coordinates [19]. GFD is invariant to rotation, scale, and mapping. This is because the Fourier coefficients are intrinsic functions that are invariant to the offset, which reflect the details of the feature point with different scales. But they have no information about its spatial position.

The wavelet transform is the result of decomposing an object into a pyramid of images with different scales and low-frequency and high-frequency components [20]. This pyramid has several necessary functions, such as simple computation, representation with different scales, constancy, spatial localization and stability, the ability to reproduce the image. The disadvantage is the sensitivity to shifting, to rotation, the lack of phase information. A survey on shape correspondence in computer vision, pattern recognition is represented in [21].

Thus, the descriptor using the spatial frequencies represents the figure of the feature point as a whole, however, without specifying its structural features. The required accuracy of the representation is achieved by choosing the number of conversion factors, which can be quite large.

2.5 Learning-based methods

The concept of the feature point is really not clear. To select the effective feature things in the image around the point under consideration, different learning-based methods are used effectively.

The supervised learning of low-dimensional feature descriptors is proposed in [22], which is applied for boosting to obtain a non-linear mapping of the input to a high-dimensional feature space. In the work [23] a learning local feature descriptor as a convex optimization problem by applying the sparsity is suggested. The proposed method can decrease the dimensionality as well as improve the descriptor effectiveness by applying the Mahalanobis norm regularization.

An evolutionary learning method is used to automatically generate the domain adaptive feature descriptor [24]. Here, the multi-objective genetic programming is applied to evolve the robust feature trees for the domain specific images with the fitness criterion taking into account the classification error rate and tree complexity.

The learning is usually used as a part of the feature extraction method. The oriented FAST and rotated BRIEF (ORB) detector detects the feature points based on the FAST algorithm. Then, a pyramid of images at different scales is created. A 9·9 window is analyzed around the feature point. Its center and orientation are calculated using the first-order moments, which are measures of some shape. The learning method is then used to increase the independence of the descriptor elements with respect to rotation. The execution of the ORB algorithm is an order of magnitude faster than the SIFT algorithm with almost the same recognition efficiency [25].

The learning-based methods don’t need manually labeled features of a point and therefore, they are more flexible than the conventional handcrafted features described above. Moreover, the trained deep learning network can select the features which capture effectively the complex morphological patterns in the image [26]. But the deep learning can only provide a limited amount of data about the feature points which are not normed and are hard to be used in the feature point descriptors.

2.6 Convolutional neural networks

The main functions of computer vision are the detection, localization, and classification of characteristic points, which can be achieved in convolutional neural networks. The Overfeat method proposes to use a multi-scale approach with adjustable windows in the convolutional neural network (CNN) [27]. In this case, the search for the feature points, ie., low-level recognition operators are performed in the first convolutional layers.

The CNN models by themselves are used for feature extraction in tools like VGGNet [28], GoogleNet [29]. They are viewed as a set of non-linear functions which are composed of a number of layers including convolution, pooling, non-linearity [30].

The calculation of the feature point search operators allows the neural work to significantly improve the image before its processing in its middle and end layers. The research in [31] offers a method of cleaning the image from noise to improve deep learning. A similar goal was achieved in [32], in which the CNN network was modified.

One of the advantages of the traditional image recognition methods is that the user can easily find evidence of how and why the methods work successfully. In CNN, this is a difficult task, because it is impossible to reasonably answer the question: what exactly has the system learned? This can be critical in some areas, such as financial, medicine image analysis, military and security systems. For reliable evidence, it is necessary to know the probable error of assessing the significance of any result. The models of CNN and machine learning are usually used as a black box, ie., they do not provide information about what exactly makes them come to their predictions [33].

2.7 Image feature detection methods comparison

The feature detector effectiveness can be tested and compared objectively only in the pattern recognition system when the proper descriptor is composed of the detector results. The descriptor is often evaluated by two tests: image matching and homography evaluation.

The image matching test is performed as follows [34]. The image collection has groups of images in each of them a reference image is and several images are distorted by scale, rotation, noise with increasing distortion (cut-off level). For all images in the collection, all descriptors are searched and calculated, which are entered into the database together with their coordinates in the image. Upon receipt of the image for recognition, the descriptors are calculated for it, and for them there are descriptors with the closest match in the database. If the distance between such descriptors is less than the threshold, then a match is recorded.

For image collections, a classification is made according to the number of matches and accuracy (average distance between descriptors). And for each algorithm of calculation of the descriptor the threshold of distance is chosen. The distance is calculated as the Euclidean distance for non-binary descriptors and as the Hamming distance for binary descriptors.

The test of the homography estimation is to calculate the coincidence of features between two images, followed by the calculation of homography. In fact, homography is the mapping of image points into a similar image, but after an affine transformation (rotation, change of perspective, etc.).

The nearest neighbor search algorithm is used to find a match to find the best match in the second image for each feature in the first image. When searching for the nearest neighbor, the Lowe rule [35] is often used, according to which the match is fixed only if the ratio of the distances from the nearest to the second nearest descriptor is less than 0.8.

So, the feature point detector is better if it provides the highest number of the feature points found in the testing image by the first method or the number of point pairs in two similar images by the second method.

The accuracy of matching is determined for the database of the image collection. The images in it are grouped by k ≤ 5 or more images of one scene but at different scales, angles, and noise levels with the increasing inability to recognize, ie., with different ranks (cut-off). Then, the accuracy is calculated as

$$P@k = \textbf{Error!}\ .$$

The frequency of the image recognition (recall) is calculated as

$$R@k\ = \textbf{Error!}\ .$$

The accuracy-frequency curves are constructed for different feature detectors. The accuracy-frequency curve of recognition tends to decrease from left to right, ie., with increasing of the cut-off rank. The method for which the curve remains higher is considered the best [36].

Due to these curves, the methods SURF, SIFT, ORB, and BRIEF have approximately equal recognition effectiveness. But the method BRIEF has the minimum complexity among them which is in two orders of magnitude less than one of SURF and SIFT [36].

The mentioned above methods analysis shows the following conclusions.

To find the feature points the whole image is scanned sequentially by the feature point detector. In many cases, the traversing is performed in many layers of the pyramid of the images with different scales. To minimize the operation number, both the pyramid forming and the second-order derivative approximation are calculated using the Gauss filtering.

The most effective detectors are based on the analysis of the brightness gradient. They provide the scale-, mapping-, and rotation-invariant detecting. They use the Harris angle detector frequently, which is complex in calculations but it finds the corners and estimates their parameters effectively. Therefore, these detectors are usually performed in software using the floating point calculations intensively.

To minimize the feature extraction, and further pattern recognition complexity, the amount of data generated by the detector is minimized preserving the information about the feature point. The methods are mostly different in the approaches which select this information. In most detectors, the luminance gradient histograms are calculated taking the informative results. The evolutionary algorithms in the learning-based methods are used for this purpose to select the effective patterns around the point centrum.

The algorithm complexity minimization contradicts both the detector effectiveness and to the high dynamic range (HDR) of the analyzed image. Therefore, the frequent methods could not recognize the feature points in the heavy luminance conditions.

3. Method for the HDR feature point detection

The HDR image pixels are distinguished in large bit width which is much higher than the eight-bit pixels of the images used in the detectors mentioned above. Such an image is effectively used in the security, military systems because it perceives the information in the difficult luminance conditions. But with the HDR image processing, the problem of the dynamic range compression of the signal has to be solved to preserve the loss of the readability of the image both in illuminated and in darkened areas.

3.1 HDR adaptive filtering

The problem of the HDR image compression is solved using the theoretical Retinex model of the scene lighting. Then, the pixel brightness is a product

$$I(x, y) = L(x, y)\cdot R(x, y),$$

where $L(x, y)$ is the illumination, and $R(x,y)$ is the reflected brightness of the object [37]. According to the Retinex approach, the image $I(x, y)$ is decomposed to the components $L(x, y)$, and $R(x, y)$. Then, the component $L(x, y)$ is processed with the dynamic range compression, and the contrast is improved in the component $R(x,y)$. These components are multiplied to obtain the resulting compressed image $I'(x,y)$. The brightness component is extracted using the function $F(I)$, which determines the illumination, so that

$$L(x,y) = F(I);\ R(x,y) = I(x,y) / F(I);$$

$$L'(x,y) = L(x,y));\ R'(x,y) = R(x,y)); \qquad (3)$$

$$I'(x,y) = L'(x,y)\cdot R'(x,y).$$

Here, the compression function (y) behaves as a logarithmical function, and the contrast improvement function y) amplifies the signal near the level of 1.0. The illumination function $F(I)$ is a smart low-pass filter (LPF) that prevents the artifacts, appearing in the resulting image, preserving the edges in it. The bilateral filter is often used as the function *F*(*I*) [38].

The bilateral filter is hard in computations. Therefore, the adaptive filter is proposed, which preserves the edges almost as well as the bilateral filter, but it is much simpler in the hardware implementation. Thanks to this, the processing of the HDR image is simplified and accelerated.

The adaptive filter structure, which preserves the edges of the image, is shown in Fig. 3. Such a filter consists of an image analyzer and an adjustable two-dimensional LPF. The idea of the Harris-Laplace detector [39] is used in the image analyzer. Such a detector is much easier to be implemented than detector (2) is and can be calculated using the fixed point computations. Its output signal is the eigenvector of the autocorrelation matrix of the neighborhood of the pixel under processing. There are five detectors $W_{|}$, $W_{-\_}$, $W_{/}$, $W_{\backslash}$, W_{*}, which are sensitive to the vertical, horizontal, inclined edges, or blobs in the image, respectively, and one LPF W_{LPF}, which estimates the local brightness of the image.

The output of such a detector is the signal in the logarithmic scale. The analyzer decision unit selects the maximum signal of the detectors and outputs it accompanying the detector number. The logarithm of the local brightness is subtracted from it providing the normalized signal *D*(*x, y*) in the high dynamic range. Fig. 4,a illustrates the results of the image analyzer for the image of the character **R**. The color of a pixel in it means the detector number.

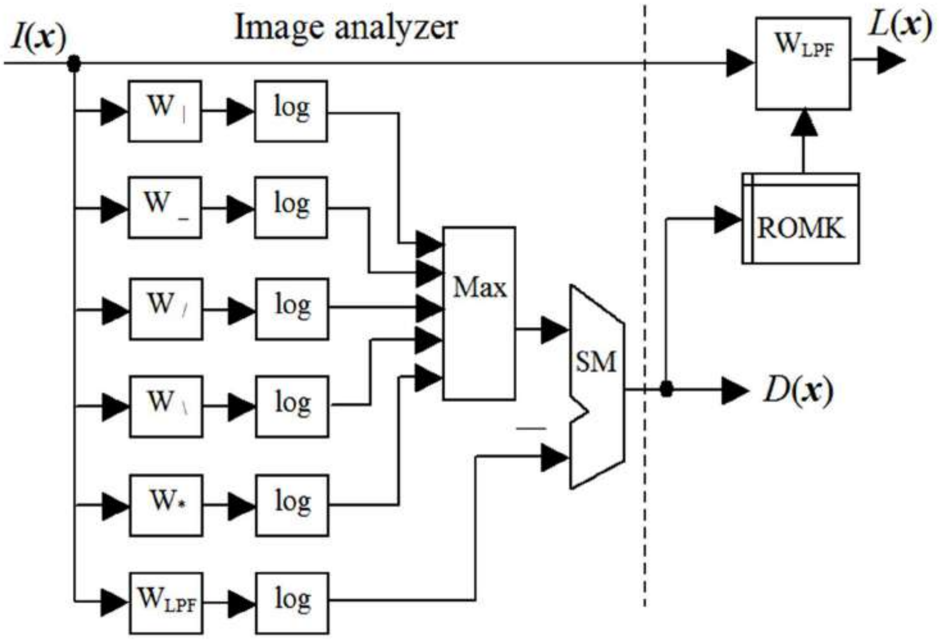


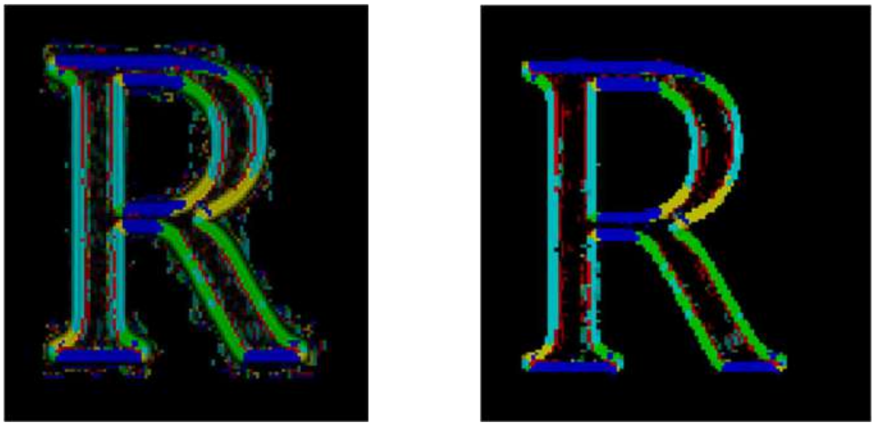
Fig. 3. Adaptive filter structure

The LPF filter contains a table ROMK of the filter kernels, which are distinguished depending on the local image type, i.e., if it is vertical, horizontal, inclined edge, and on its strength. The output signal of the image analyzer selects the proper kernel in the table for every pixel in the image. As a result, the image is filtered, providing the sharp edges in any luminance conditions. The resulting adaptive filter computes the illumination function *F*(*I*) and is less complex than the bilateral filter. The work [40] describes this filter hardware implementation in detail.

3.2 Feature point extraction

To search for the feature points in the image, two additional computational steps are added to the adaptive HDR filter described above. The first one is the noise filtering step. Because of the logarithmical scale of the image *D*(*x, y*), the usual linear filtering methods are not fitted. In this situation, the maximum homogeneity neighbor (MHN) filter is used [41]. The MHN filter is simplified here because the pixels *D*(*x, y*) have the small bit width and are distinguished in five different colors, as in Fig. 4. The pixel *D*(*x, y*) is smoothed in the filter if a stencil is found, which covers the pixels of the same color. Otherwise, this pixel gets the background color. The filtered image is shown in Fig. 4, *b*.

In the second step, the feature points are found. These points are blobs, corners in the edges, corners in the lines, intersections of the lines and edges. The examples of them are shown in Fig 5. These feature points are searched using the MHN method too. But the spatial stencils are adapted to the considered feature. For example, if the corner is considered, it has the proper angle and direction in the space which are coded by the colors. The selected feature point has the coordinates (*x, y*) of the respective feature filter with the maximum output magnitude in the considered image locality. For such behavior, this detector is rather similar to one in the FAST and BRISC methods.



a) b)

Fig.4. The output of the adaptive filter (a) and after the noise filtering (b)

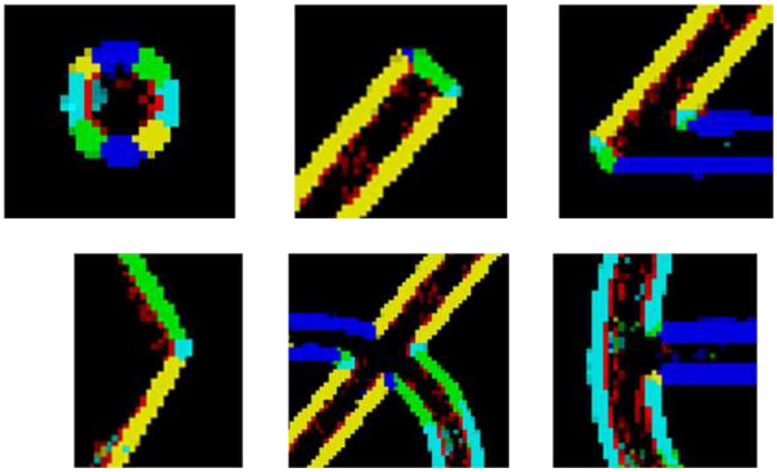


Fig.5. Examples of the feature point localizations

To do the scalable feature point searching, the derived logarithmic image has to be decimated many times. Such a decimation is performed using the MHN method except that it is performed to each even pixel. By this process, the feature point coordinates coincide with each other in the images with different scales.

The next step of the feature point descriptor forming consists in calculating the distribution diagrams as it is shown in [42].

3.3. Experimental results and future work

At present, the artificial intelligence engine is under development, which is implemented in FPGA of medium or large size. The input data is an HDR video camera data flow. To process the image with the HDR compression, the video camera was designed on the base of the Lattice HDR-60 board, which utilizes ECP3-70 FPGA. As a video sensor, the Aptina MT9M024 chip is used, which produces the 720·1280 HDR image stream at a rate of 60 frames per second with a dynamic range of 120 dB.

The colored image channel is split into the brightness and color channels. The brightness signal with the 20-bit pixels is compressed to 8-bit pixels due to the equations (3). Then, the colored image is restored using the color channel. The image analyzer filters (Fig. 3) are implemented as the pipelined multiplier-free adder networks. These networks provide both high-speed computations for the multi-bit data and small hardware volume comparing to the network of the hardware multipliers. The functions (y) and y) in (3) are implemented using the piecewise linear interpolation.

Fig. 6 illustrates the scene image with the heavy luminance conditions which is compressed by the video camera and the respective image for the feature extracton. It shows that the feature points can be selected robustly in such strong conditions.

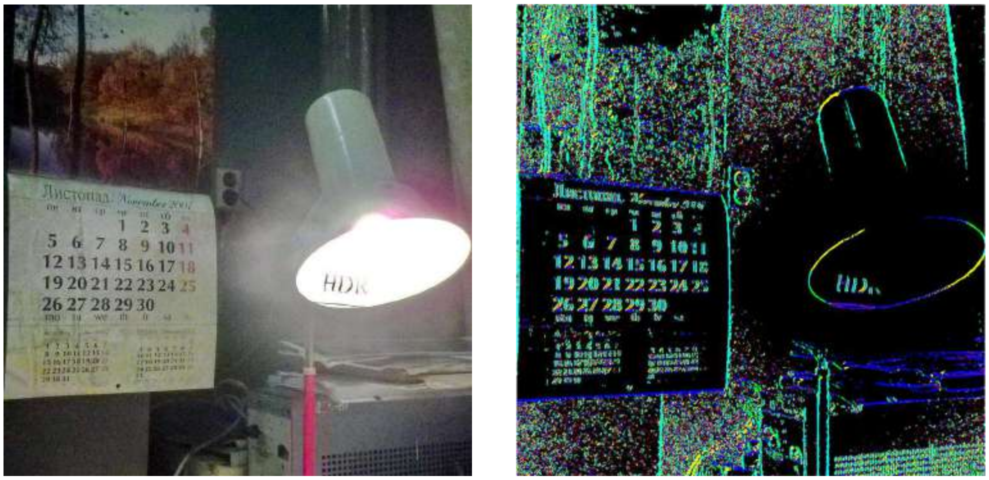


Fig.6. HDR compressed image and results of its processing by the analyzer

This system was probed to analyze the blood image which was taken from the repository of the information on the biochemicals and cells in blood and body fluids [43]. Fig. 7 illustrates the results of the use of the developed system. Its analysis shows that the system can reliably select the cells and represent its features, which helps to recognize and make classification of the unnormal cells. Such cells have a set of feature points representing the angles which form a ring.

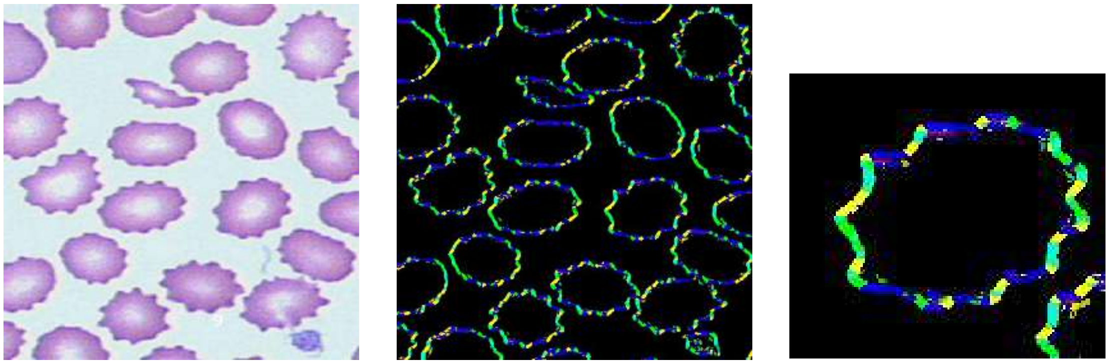


Fig.7. Initial image and results of its processing by the analyzer

The future work consists of development of the pattern recognition system. The pattern recognition process has two distinguished stages. In the first stage, the image is transformed into a pyramid of feature frames. For this process, an adaptive filter is used as in Fig. 3, which both compresses the pixel dynamic range and preserves the edges of the image. But for the noise filtering, and for the frame pyramid forming the MHN filter is used.

In the second stage, the feature points are found. These points are blobs, corners in the edges, corners in the lines, intersections of the lines, and edges. These feature points are searched using the MHN method as well. Then, the feature descriptors are formed as like as in the SIFT method.

At the period of learning, the computed feature descriptors are stored in the database, and at the working period, the found feature descriptors are compared to ones in the database.

To design the pipelined datapath in a short time, a framework is designed which helps to compile the data path algorithm representation into the FPGA hardware described in VHDL. The algorithm can be represented in the TensorFlow language as well.

4. Conclusions

A set of different methods of feature point extraction is analyzed. This analysis shows that the most effective detectors are based on the analysis of the brightness gradient. They usually use the Harris angle detector, which is complex in calculations. The algorithm complexity minimization contradicts both the detector effectiveness and the high dynamic range of the analyzed image. Therefore, the high-peed methods could not recognize the feature points in the heavy luminance conditions.

The modification of the HDR image compression algorithm based on the Retinex method is proposed. It contains an adaptive filter, which preserves the image edges. The filter is based on a set of feature detectors performing the Harris-Laplace transform which is much simpler than the Harris angle detector. Such a filter reduces the compression complexity. A prototype of the HDR video camera is designed which provides sharp images both in illuminated and in darkened areas of the scene. The resulting signal of this camera simplifies the design of the pattern recognition system. The next step will be the design of the artificial intelligence engine, which is implemented in FPGA of medium or large size.

References

[1] M. Aguado, *Feature Extraction and Image Processing For Computer Vision.* S.L.: Elsevier Academic Press, 2019.

[2] T. Tuytelaars and K. Mikolajczyk, “Local Invariant Feature Detectors: A Survey,” *Foundations and Trends® in Computer Graphics and Vision*, vol. 3, no. 3, pp. 177–280, 2008, doi: https://doi.org/10.1561/0600000017.

[3] S. Krig, “Interest Point Detector and Feature Descriptor Survey,” in *Computer Vision Metrics: Survey, Taxonomy, and Analysis*, Berkeley, CA: Apress, 2014, pp. 217–282. doi: https://doi.org/10.1007/9781430259305_6.

[4] P. Kovesi, “Image Features from Phase Congruency,” *MIT Press Journals*, vol. 1, no. 3, pp. 1–27, Jan. 1995.

[5] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models,” *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988, doi: https://doi.org/10.1007/BF00133570.

[6] H. Moravec, “Towards Automatic Visual Obstacle Avoidance,” in *Proceedings of 5th International Joint Conference on Artificial Intelligence*, 1977, p. 584.

[7] C. Harris and M. Stephens, “A Combined Corner and Edge Detector,” in *Proceedings of Fourth Alvey Vision Conference*, 1988, pp. 147–151.

[8] D. G. Lowe, “Distinctive Image Features from ScaleInvariant Keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004, doi: https://doi.org/10.1023/B:VISI.0000029664.99615.94.

[9] H. Bay, A. Ess, T. Tuytelaars, and V. Gool, “SpeededUp Robust Features (SURF),” *Similarity Matching in Computer Vision and Multimedia*, vol. 110, no. 3, pp. 346–359, 2008, doi: https://doi.org/10.1016/j.cviu.2007.09.014.

[10] D. Weng, Y. Wang, M. Gong, D. Tao, H. Wei, and D. Huang, “DERF: Distinctive Efficient Robust Features from the Biological Modeling of the P Ganglion Cells,” *IEEE Transactions on Image Processing*, vol. 24, no. 8, pp. 2287–2302, doi: https://doi.org/10.1109/TIP.2015.2409739.

[11] J.-M. Morel and G. Yu, “ASIFT: A New Framework for Fully Affine Invariant Image Comparison,” *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 438–469, Jan. 2009, doi: https://doi.org/10.1137/080732730.

[12] E. Tola, V. Lepetit, and P. Fua, “DAISY: An Efficient Dense Descriptor Applied to WideBaseline Stereo,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 815–830, doi: https://doi.org/10.1109/TPAMI.2009.77.

[13] F. Tombari, A. Franchi, and L. Di, “BOLD Features to Detect Textureless Objects,” in *2013 IEEE International Conference on Computer Vision*, pp. 1265–1272. doi: https://doi.org/10.1109/ICCV.2013.160.

[14] E. Rosten and T. Drummond, “Machine Learning for HighSpeed Corner Detection,” in *Computer Vision – ECCV 2006*, A. Leonardis, H. Bischof, and A. Pinz, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 430–443.

[15] E. Mair, G. D. Hager, D. Burschka, M. Suppa, and G. Hirzinger, “Adaptive and Generic Corner Detection Based on the Accelerated Segment Test,” in *Computer Vision – ECCV 2010*, K. Daniilidis, P. Maragos, and N. Paragios, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 183–196.

[16] S. Leutenegger, M. Chli, and R. Y. Siegwart, “BRISK: Binary Robust invariant scalable keypoints,” in *2011 International Conference on Computer Vision*, pp. 2548–2555. doi: https://doi.org/10.1109/ICCV.2011.6126542.

[17] A. Alahi, R. Ortiz, and P. Vandergheynst, “FREAK: Fast Retina Keypoint,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 510–517. doi: https://doi.org/10.1109/CVPR.2012.6247715.

[18] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua, “BRIEF: Computing a Local Binary Descriptor Very Fast,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1281–1298, doi: https://doi.org/10.1109/TPAMI.2011.222.

[19] D. Zhang and G. Lu, “Generic Fourier descriptor for shapebased image retrieval,” in *Proceedings. IEEE International Conference on Multimedia and Expo*, pp. 425–428 vol.1. doi: https://doi.org/10.1109/ICME.2002.1035809.

[20] Nabout, Adnan Abou and B. Tibken, “Wavelet Descriptors for Object Recognition Using Mexican Hat Function,” *16th IFAC World Congress*, vol. 38, no. 1, pp. 1107–1112, 2005, doi: https://doi.org/10.3182/200507036CZ1902.00186.

[21] O. van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or, “A Survey on Shape Correspondence,” *Computer Graphics Forum*, vol. 30, no. 6, pp. 1681–1707, Jul. 2011, doi: https://doi.org/10.1111/j.1467-8659.2011.01884.x.

[22] T. Trzcinski, M. Christoudias, and V. Lepetit, “Learning Image Descriptors with Boosting,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 597–610, doi: https://doi.org/10.1109/TPAMI.2014.2343961.

[23] K. Simonyan, A. Vedaldi, and A. Zisserman, “Learning Local Feature Descriptors Using Convex Optimisation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 8, pp. 1573–1585, doi: https://doi.org/10.1109/TPAMI.2014.2301163.

[24] L. Shao, L. Liu, and X. Li, “Feature Learning for Image Classification Via Multiobjective Genetic Programming,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 7, pp. 1359–1371, doi: https://doi.org/10.1109/TNNLS.2013.2293418.

[25] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: An efficient alternative to SIFT or SURF,” in *2011 International Conference on Computer Vision*, pp. 2564–2571. doi: https://doi.org/10.1109/ICCV.2011.6126544.

[26] G. Wu, M. Kim, Q. Wang, B. C. Munsell, and D. Shen, “Scalable HighPerformance Image Registration Framework by Unsupervised Deep Feature Representations Learning,” *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 7, pp. 1505–1516, doi: https://doi.org/10.1109/TBME.2015.2496253.

[27] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Yann LeCun, "OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks," Dec. 2013, doi: https://doi.org/10.48550/arxiv.1312.6229.

[28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations (ICLR 2015)*, Computational and Biological Learning Society, 2015, pp. 1–14.

[29] C. Szegedy *et al.*, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9. doi: https://doi.org/10.1109/CVPR.2015.7298594.

[30] L. Zheng, Y. Yang, and Q. Tian, "SIFT Meets CNN: A Decade Survey of Instance Retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 5, pp. 1224–1244, doi: https://doi.org/10.1109/TPAMI.2017.2709749.

[31] K. Gul and B. K. Gunturk, "Spatial and Angular Resolution Enhancement of Light Fields Using Convolutional Neural Networks," *IEEE Transactions on Image Processing*, vol. 27, no. 5, pp. 2146–2159, doi: https://doi.org/10.1109/TIP.2018.2794181.

[32] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, doi: https://doi.org/10.1109/TIP.2017.2662206.

[33] Q. Zhang and S. Zhu, "Visual interpretability for deep learning: a survey," *Frontiers of Information Technology & Electronic Engineering*, vol. 19, no. 1, pp. 27–39, 2018, doi: https://doi.org/10.1631/FITEE.1700808.

[34] M. Muja and D. G. Lowe, "Scalable Nearest Neighbor Algorithms for High Dimensional Data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 11, pp. 2227–2240, doi: https://doi.org/10.1109/TPAMI.2014.2321376.

[35] D. G. Lowe, "Distinctive Image Features from ScaleInvariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004, doi: https://doi.org/10.1023/B:VISI.0000029664.99615.94.

[36] N. Khan, B. McCane, and S. Mills, "Better than SIFT?" *Machine Vision and Applications*, vol. 26, no. 6, pp. 819–836, 2015, doi: https://doi.org/10.1007/s0013801506897.

[37] E. H. Land and J. J. McCann, "Lightness and Retinex Theory," *Journal of the Optical Society of America*, vol. 61, no. 1, Art. no. 1, 1971, doi: https://doi.org/10.1364/JOSA.61.000001.

[38] S. Paris, P. Kornprobst, J. Tumblin, and F. Durand, *Bilateral Filtering: Theory and Applications*. now, 2009, pp. 1-. Available: http://ieeexplore.ieee.org/document/8187212

[39] M. Hassaballah, Abdelmgeid, Aly Amin, and Alshazly, Hammam A, "Image Features Detection, Description and Matching," in *Image Feature Detectors and Descriptors: Foundations and Applications*, A. I. Awad and M. Hassaballah, Eds., Cham: Springer International Publishing, 2016, pp. 11–45. doi: https://doi.org/10.1007/9783319288543_2.

[40] A. Sergiyenko, P. Serhiienko, and J. Zorin, "High Dynamic Range Video Camera with Elements of the Pattern Recognition," in *2018 IEEE 38th International Conference on Electronics and Nanotechnology (ELNANO)*, pp. 435–438. doi: https://doi.org/10.1109/ELNANO.2018.8477556.

[41] M. Nagao and T. Matsuyama, "Edge preserving smoothing," *Computer Graphics and Image Processing*, vol. 9, no. 4, pp. 394–407, 1979, doi: https://doi.org/10.1016/0146664X(79)901023.

[42] A. Sergiyenko, P. Serhiienko, M. Orlova, and O. Molchanov, "System of Feature Extraction for Video Pattern Recognition on FPGA," in *2019 IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON)*, pp. 1175–1178. doi: https://doi.org/10.1109/UKRCON.2019.8879958.

[43] "Medical Laboratories Portal," *Medical-labs.net*. http://www.medical-labs.net/wp-content/uploads/2014/01/Crenated-Cells.jpg

UDC 004.383
DOI: https://doi.org/10.20535/2708-4930.2.2021.244189

I. MOZGHOVYI
A. SERGIYENKO
R. YERSHOV

GIF IMAGE HARDWARE COMPRESSORS

Increasing requirements for data transfer and storage is one of the crucial questions now. There are several ways of high-speed data transmission, but they meet limited requirements applied to their narrowly focused specific target. The data compression approach gives the solution to the problems of high-speed transfer and low-volume data storage. This paper is devoted to the compression of GIF images, using a modified LZW algorithm with a tree-based dictionary. It has led to a decrease in lookup time and an increase in the speed of data compression, and in turn, allows developing the method of constructing a hardware compression accelerator during the future research.

Keywords: FPGA, GIF, lossless compression, image compression, dictionary, hardware acceleration

1. Introduction

Nowadays, the problem of data transferring optimization is becoming one of the most significant. Whereas the size of data increases, there should be a way to transfer it with the highest speed. A solution to this problem depends on the branch of its application. There is a list of the solutions shown in Fig. 1.

One of them is the parallel busses. They are used mostly for:

- Peripheral connections to the computer motherboard (e.g. PCI express bus for connection of GPU module);
- System on chip interconnection busses (e.g. Avalon interface for connection of Intel FPGA modules);
- Standardized system busses for microcontrollers (e.g. AHP APB busses of ARM ® Cortex ® processors).

Another approach is the high-speed serial interfaces. An application of it can be found in:

- Network interfaces;
- The connections between modules on a single board;
- Data transmission for high-speed ADC modules;
- Low-voltage differential signaling [14, 15].

Despite the different areas of application, these solutions have a set of common problems. The first one is that they are used only for data transmission**.** As a result, they cannot solve the problem of data storage, which is also important. The next one is a narrowly focused area of application. It means that each solution has a specific target, which is non-scalable to another one.

It is well known, that the use of the data compression can be found in more branches than for the high-speed data transfer. In addition, a combination of high-speed interfaces and data compression is a good practice. For example, in the latest versions of the HDMI interface, the highest data rates are possible only using the Display Stream Compression mode [12, 13].

The data compression is used not only for data transfers but also for storage. Therefore, developing an efficient approach of data compression solves not only the transmission problem but also the storage. It does not matter if it is big data storage or just a memory block to keep the buffered image. Decreasing the size of a single file optimizes both cases.

The main point is that usage of the different formats can modify the entire image corresponding to the specific algorithm. In most cases, it is a compression method algorithm. The main difference between all compression methods is that if it is lossless or not. The term of the "information loss" means that some compression methods cannot guarantee exactly the same image after its decompression. It will definitely differ from the original image. However, in some cases, a human eye cannot notice the difference between the source and decompressed image, and using the method with losses is acceptable. Mostly they are used for multimedia, where little distortion after decompression is insufficient. In addition, there are other features of compression methods, e.g. dictionary or run-length compression, compression ratio, compression speed [7], etc.

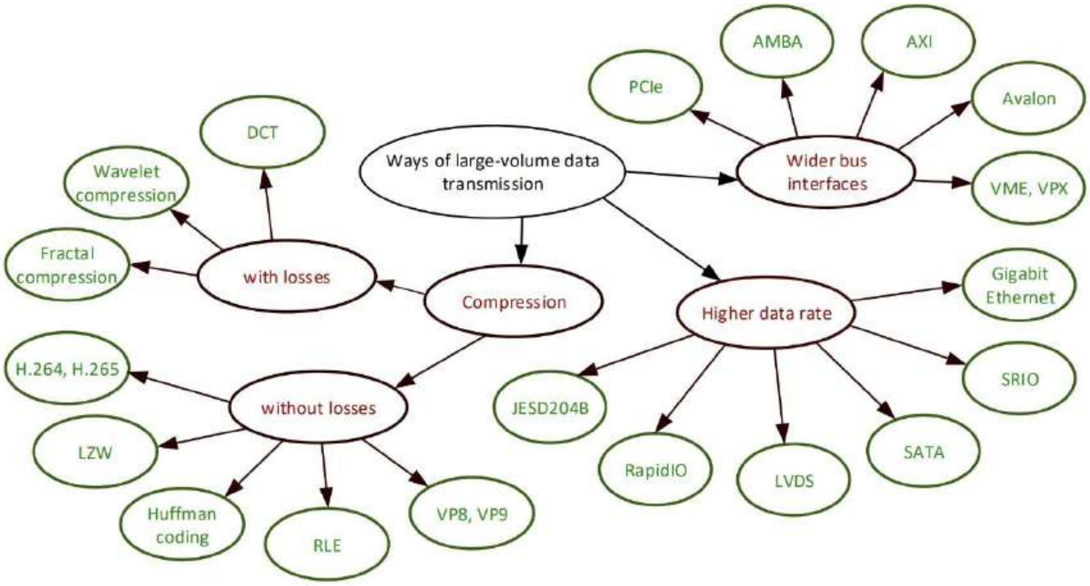


Fig. 1. Data transmission solutions

The main point is that usage of the different formats can modify the entire image corresponding to the specific algorithm. In most cases, it is a compression method algorithm. The main difference between all compression methods is that if it is lossless or not. The term of the "information loss" means that some compression methods cannot guarantee exactly the same image after its decompression. It will definitely differ from the original image. However, in some cases, a human eye cannot notice the difference between the source and decompressed image, and using the method with losses is acceptable. Mostly they are used for multimedia, where little distortion after decompression is insufficient. In addition, there are other features of compression methods, e.g. dictionary or run-length compression, compression ratio, compression speed [7], etc.

Nevertheless, the lossless compression is obligatory for the original photo, medical image, or document image storage. For this purposes the compression programms are usually used. But the hardware compressor could both speed-up the compression, decrease the energy consumption, and probably, decrease the stored file volume.

This article considers the questions of the lossless compression algorithm selection and its hardware implementation issues.

2. Selection of a compression method

There are some points about the selection of the compression method. In general, all the lossless compression methods are divided into:

- Run-Length Encoding;
- Statistical Methods;
- Dictionary methods.

All of these classifications find their implementations in software, but the question of their hardware implementation remains actual. In the comparative table of the compression methods in [1] it was mentioned that the run-length encoding methods do not give a good compression ratio, therefore, they are omitted, despite their losslessness. Only two variants are left to choose from: the statistical method or the dictionary method. The general scheme of a compression method consists of two parts: the model and the coder. The model finds the redundancy in the input data and sends it to the coder, which replaces the repetitive fragments with the corresponding codes. But there is a difference between statistical and dictionary methods, so the short overview should be provided to clear the details.

2.1 Statistical methods

There is a clean separation onto model and coder parts. The statistical model assigns the values to the events (some data fragment found) depending on the probability of their appearance in the input data sequence. The more frequency of the event occurrence the more the value. The main problem of a hardware implementation of a statistical method is that they are mostly based on the Markov stochastic modeling. In the paper [4] it was mentioned that the compression ratio of the statistical methods is limited by the usage of the multi-symbol alphabet zeroth-order modeling. On the other hand, the speed of the compression is limited by the use of the binary alphabets in the high-order modeling. The author’s explanation of it is in that the data in the binary symbol alphabet, only a few bits are processed in each cycle. Finally, the paper [4] represents the next features of the hardware implementation:

- Hardware complexity of the zeroth-order modeling and not impressive productivity results.
- High-order modeling does not give good performance characteristics. They are not comparable with the results of the dictionary methods.
- Tree-based implementations using Huffman coding showed better results but the problem remains of adaptation to the difference in input image sequence. In addition, the best performance was achieved only using the content-addressed memory. And the best-mentioned compression ratio of 0.5 is also not impressive.

2.2 Dictionary methods

In contradiction to the statistical methods, the dictionary methods provide compression using a special dictionary which is pre-determined, or filled during the input data processing. Such a technique does not use a statistical model or variable-sized codes. Each subsequence of bits from the input data stream is represented as a token, or a record, in a compression dictionary.

Depending on the method, the dictionary may be static or dynamic. The static one is filled before the compression process is started and the repeated blocks of data are replaced only in case when they are available in the pre-determined dictionary. The dynamic dictionary is adaptive and is partially filled at the step of initialization and then appended with the new records during the data processing.

When the dictionary is set up, the compression of input data is performed in a way of replacing the repeated portions of bits, strictly according to the dictionary table. In addition, such compressors are not narrowly focused on target data format, and may be used for general purposes. They can compress the audio data as well as text, what makes them popular.

In a case of implementation of a compressor with a wide application spectrum, the dynamic dictionary is more suitable than the static one [7]. Furthermore, in accordance to the principle of the data processing, there is an assumption that the dictionary compressing methods are more suitable for the hardware implementation.

The paper [4] also supports the hypothesis that dictionary compression methods are better to be implemented in hardware then the statistical methods. First of all, they are due to achieving good throughput and the competitive compression ratio. In addition, these methods are good for compression of non-streaming data, what widens its area of application. So, to clear up the benefits of using the dictionary compression methods for hardware implementation some examples should be provided.

All four software and hardware examples, described in [4] use the derivatives methods from the Lempel-Ziv-1 (LZ1) algorithm. As a first example an ALDC algorithm was represented which is implemented in a 0.8-um CMOS technology and clocked at 40 MHz obtaining a throughput of 320 Mb/s. This algorithm was developed by IBM which is used in utilities like Pkzip and ARJ. The implementation of AHA coprocessor gives a performance of the same 320 Mb/s with the 40MHz clock frequency but in the 0.5-um CMOS technology.

The next example is the STAC/Hifn device, representing the LZS algorithm. It was implemented in a 0.35-µm CMOS technology, was clocked at 80 MHz and showed a throughput of 640 Mb/s. This device consists of a full-duplex architecture meaning that it can compress and decompress the data simultaneously. Both of these chips use the CAM memory to store the dictionary and enable the parallel searching and adaptation.

Another example of the hardware implementation of a dictionary-based method is a PE-based processing element architecture for LZ1 algorithm. With the constant data input rate the post-layout simulation showed a performance of 700Mb/s in the 0.5-um CMOS technology. However, this implementation is applicable only for compressing the ASCII coded models due to the 7-bit basic symbol width.

In comparison to the dictionary methods, the statistical methods showed worse performance characteristics. The same paper [4] describes an example of a chip representing a tenth-order Markov model with the associated binary arithmetic coder, which is implemented in a 0.8 µm CMOS technology and is clocked at 25 MHz. Its compression ratio is in the order of 0.5, while the speed is data dependent but typically is around not impressing 3Mbit/s.

The examples with the Huffman coding technology showed better performance, but worse than LZ1-based ones did. The first one showed 95.2 Mb/s for compression and 60.6 Mb/s for decompression in a 2-µm SCMOS technology with a clocking frequency of 83.3MHz. To achieve this result a CAM memory modules were used to speed up the tree adaptation process.

3. LZW compression method

The goal of the current research is to find a way to improve the existing GIF [5] image format. The main benefit of using GIF is that the image compression is provided using LZW [8] dictionary lossless compression method. In some cases, it is necessary to keep the image as it was before the compression. For example, the image of schematics with small notations or values. In addition, a GIF file can be represented as an animation, due to the compressed sequence of image frames inside a file [5]. Different solutions can be found to improve the existing GIF image format. Generally, they can be divided into the optimization of the color table and improving the LZW compression method. Our research is about the modification of the LZW compression method.

Some research has addressed the problem of its hardware implementation. The authors of the paper [9] propose an FPGA-based implementation of the LZW algorithm. The main architectural feature of this FPGA implementation is an FPGA-suitable hash table that consists of buckets each of which is composed of 8 entities. Each entity stores a 12-bit pointer, 8-bit character, and 12-bit back pointer. The data table is divided into 8 parts what facilitates the reading of 8 values at one time. Having a back pointer in a record makes easier the search of included values without checking eight entities in the bucket one by one. Also, there is a 4-bit value in the bucket record which can easily determine if the element is already stored. For this hash table was used three operations: initialization, search, and adding.

As for the hardware aspects, in order to implement that hash table was used block RAMs, configured in dual-port mode. The total amount is eighteen 18 Kb block RAMs. The final device was implemented on a circuit with the Xilinx Virtex-7 FPGA. Implementation of 1 instance will take:

- 104 (0.02% of available on FPGA) Slice registers;
- 346 (0.11%) Slice LUTs;
- 18 (0.87%) 18K block RAMs;
- The clock frequency is 179.99 MHz.

The experimental results showed not a big difference with sequential LZW compression on the Intel Core i7-4790 with a 3.6 GHz clock frequency. If the test image has more common regions the sequential implementation is even faster. Testing one image the FPGA speed-up factor was 0.34:1 over the CPU. But if we take a look at the percentage of used hardware components, only a small part of them were actually active. If the implementation consisted of 24 circuits, the hardware parameters are the next:

- 3120 (0.51%) Slice registers;
- 7782 (2.56%) Slice LUTs;
- 432 (20.97%) 18K block RAMs;
- The clock frequency is 163.35 MHz.

However, the maximum clock frequency decreases with the growth of instances the results showed that such a solution gives a speed factor up to 23.51 over the sequential implementation on the CPU [9].

There is another study [3], proposing to use the custom compression method that implements a bit plane slicing and adaptive Huffman encoding for the LZW dictionary. This approach gives a result of a higher compression ratio up by 2 times more than the original method. One more way [10] is to improve the utilization of the dictionary by dividing it into sets. This allows decreasing the lookup time and partially operating in a parallel way. Combining all of the recommended methods, the own FPGA implementation can be designed. Hardware implementation can find its application in different branches, e.g. space technologies [11].

To obtain an efficient implementation of a hardware compressor, the answers to 3 questions should be found:

- What might be pipelined and parallelized and in what way?
- What processing stages depend on the results of the previous ones?
- What parts of the algorithm might be scalable?

4. Implementation aspects

The proposed method of the hardware compressor implementation includes both hardware and software parts. For today, several companies (e.g. Intel, Xilinx) have suggested a solution to such implementation using the technology of the “System-on-Chip” (SoC). For example, Intel has a family of FPGAs Cyclone® V SoC, which implements an FPGA and an ARM dual-core processor ARM® Cortex®-A9 on a single chip. The communication between hardware and software subsystems is performed using the hardware processor system IP Core, which allows interconnects FPGA interfaces with the ARM processing core [16]. Fig. 2 represents the scheme, where can be seen the connections between each part of the system. Other aspects of implementation give answers to 3 questions from the review section.

Firstly, the simplified structure of the GIF file shown in Fig. 3 should be analyzed. There are many things, which can be modified to get higher performance, but in our case, it should be focused on the fact that GIF format supports displaying a sequence of images as frames. Therefore, it should be considered that pipelining and parallelization might be applied either to the image regions (after dividing the image into regions) or to each image from the distributed sequence into each processing branch if the sequence of frames is processed. For example, if 4 instances of a hardware accelerator are available, each image on each processor for the frame sequence can be distributed. And, properly to the sequence, the compressed frames are added to the result file.

Another point is that the LZW algorithm mostly consists of sequential processes. The first step of the algorithm is to initialize the first 255 dictionary records with default values from 0 to 255. This step cannot be parallelized for obvious reasons. To decrease the lookup time of occasion search, the modified tree-based structure of the compression dictionary can be used. Each record of this dictionary consists of the fields shown in table 1.

The address represents the actual offset in memory (RAM) to access the byte value from the dictionary default values or the ancestors. It is similar to the pointer in the C programming language. This field is necessary for getting access to the proper nodes.

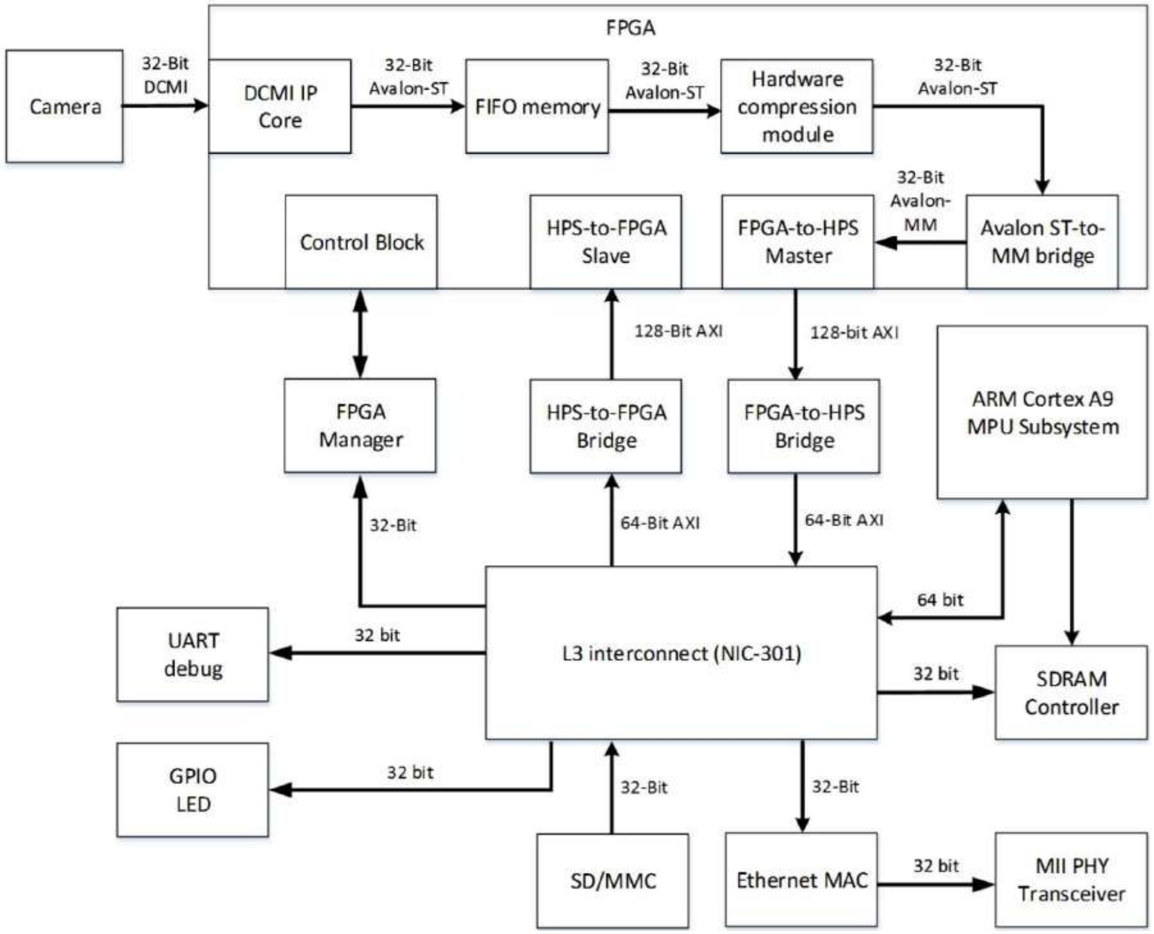


Fig. 2. Final Device Scheme

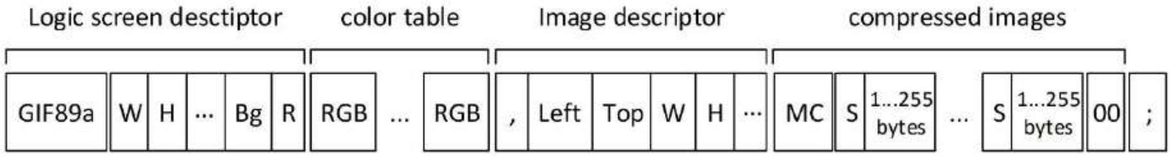


Fig. 3. GIF File Structure

Table 1

Tree-based dictionary nodes

	Nodes		
Address	97	268	297
Value	(-: «a»)	(97: «b»)	(266: «c»)

The Value is the combination of the ancestor's address and the default record. For example, if we have the text string "ABC", the node will have the next structure:

Address 297 is the newly assigned address of the node. 268 is the address of the ancestor, which has the address 268 and the value 97: 268, where 97 is the address of its ancestor - default record "a".

Another point is to choose the correct form of a tree structure. We decided to use the AVL [18] structure because it is balanced, and the balancing process is performed on a step of adding a new node.

The scaling might be applied to different features. For example, service data of GIF file allow configuration of such parameters:

- Number of bits per color;
- Amount of frames(images) per file;
- Image resolution.

However, the main scaling parameter, in our case, is that an FPGA allows multiple instance implementations. The scaling of this parameter is limited only to the amount of the FPGA components.

5. Discussion

Summarizing the above study, the developed hardware compression unit does not show breakthrough characteristics. However, during analyzing the FPGA resources that are actually used [16], it can be seen that it has enough space to implement at least 20 instances of the hardware part of the investigated compression unit. Moreover, it is not the highest-performance Intel FPGA, which can be offered. The leading Intel FPGAs Stratix 10, which has many more resources than any Cyclone V FPGA, allows increasing the performance characteristics by several times. Assuming the above, the limitations of the software embodiments of the image compressors can be overcome.

Another benefit of an FPGA using is that a low-power consumption feature can be achieved under changing of some parameters of the synthesis constraints files (*.ucf). Therefore, one of the aspects of future research can also be dedicated to developing the image compression unit embodiment which is optimized by the hardware volume and low-power consumption criteria.

6. Conclusions

This paper describes, in general, the scientific problem of data transmission, and what benefits data compression gives, comparing to other solutions. To choose the correct way of further research, all the factors were cleared up and given some proves about choice of compression method to implement. The main benefit of using dictionary compression methods against statistical was the compression speed. As a compression method to review, the LZW algorithm was chosen for several reasons. The first one is that the LZW method performs data compression without losses. In addition, this compression method is used in GIF files compression, so it seems that it is good for both data and image compression.

To have an efficient implementation, some aspects were discussed: what parts of the algorithm implementation should be parallelized and pipelined; what processing stages depend on the results of the previous ones; what parameters might be scalable. To show a possible implementation as a final device was shown the scheme with all the main processing units and interconnection between them. It was described as a device, implemented using the technology of “System-on-Chip” which represents a complex device of an FPGA with the ARM processor on a single chip. This technology is widely used nowadays which makes it possible to use the ready-made drivers and solutions to implement the compressor as a final device, so facilitate its development. In the discussion section, the possible ways of the future research were presented.

References

[1] M. Sharma, “Compression Using Huffman Coding,” *JCSNS International Journal of Computer Science and Network Security*, vol. 10, no. 5, May 2010.

[2] H. Rubaiyat, “Data Compression using Huffman based LZW Encoding Technique,” *International Journal of Scientific & Engineering Research*, no. 11, 2011.

[3] A. Taleb, H. Mustafa, A. Khtoom, and I. Gharaibeh, “Improving LZW image compression,” *European Journal of Scientific Research*, vol. 44, no. 3, pp. 502–509, Aug. 2010.

[4] K. Papadopoulos and I. Papaefstathiou, “TitanR: A Reconfigurable Hardware Implementation of a HighSpeed Compressor,” in *2008 16th International Symposium on FieldProgrammable Custom Computing Machines*, pp. 216–225. doi: https://doi.org/10.1109/FCCM.2008.14.

[5] CompuServe Incorporated, “CompuServe Graphics Interchange Format (GIF),” 1987

[6] J. Miano, *Compressed image file formats: JPEG, PNG, GIF, XBM, BMP*. New York: Addison-Wesley, 1999.

[7] D. Salomon, *Data Compression*. London: Springer London, 2007. doi: https://doi.org/10.1007/978-1-84628-603-2.

[8] T. Welch, “A Technique for High-Performance Data Compression,” *Computer*, vol. 17, no. 6, pp. 8–19, Jun. 1984, doi: https://doi.org/10.1109/mc.1984.1659158.

[9] X. Zhou, Y. Ito, and K. Nakano, “An Efficient Implementation of LZW Compression in the FPGA,” in *International Conference on Algorithms and Architectures for Parallel Processing*, Jan. 2016, pp. 512–520. doi: https://doi.org/10.1007/978-3-319-49583-5_39.

[10] W. Cui, “New LZW data compression algorithm and its FPGA implementation,” in *Picture Coding Symposium*, Jan. 2007.

[11] P.-S. Yeh, “Implementation of CCSDS Lossless Data Compression for Space and Data Archive Applications,” *SpaceOps 2002 Conference*, Mar. 2002, doi: https://doi.org/10.2514/6.2002-t5-12.

[12] “HDMI 2.1 Overview,” *HDMI Forum, Inc*, 2017. hdmi.org (accessed Jan. 10, 2017).

[13] “HDMI 2.1 Press Release,” *HDMI Forum, Inc*, 2017. hdmi.org (accessed Jan. 10, 2017).

[14] P. Heydari, “Design and analysis of lowvoltage currentmode logic buffers,” in *Fourth International Symposium on Quality Electronic Design, 2003. Proceedings.*, pp. 293–298. doi: https://doi.org/10.1109/ISQED.2003.1194748.

[15] A. Boni, A. Pierazzi, and D. Vecchi, “LVDS I/O interface for Gb/sperpin operation in 0.35/spl mu/m CMOS,” *IEEE Journal of SolidState Circuits*, vol. 36, no. 4, pp. 706–711, doi: https://doi.org/10.1109/4.913751.

[16] “Cyclone V Hard Processor System Technical Reference Manual,” *Intel*, 2018. https://www.intel.com/content/www/us/en/docs/programmable/683126/17-1/introduction.html

[17] “DE0-Nano-SoC User Manual,” Terasic Inc, 2019. Available: https://www.terasic.com.tw/attachment/archive/941/DE0-Nano-SoC_User_manual_rev.D0.pdf

[18] M. Adelson-Velskii, “An algorithm for organization of information,” *Doklady Akademii Nauk SSSR*, pp. 263–266, 1962.

UDC 004.414.22
DOI: https://doi.org/10.20535/2708-4930.2.2021.247770

I. KLYMENKO
Y. BUTSKYI
K. HRYSHCHENKO
M. SIVACHENKO
V. KRYVETS V
D. KRYVOSHEI
T. NGUEN

ARCHITECTURAL REVIEW AND CONCEPTUAL DEVELOPMENT OF FACULTY INFORMATION SYSTEM "KPI-CONNECT"

This paper is dedicated to development model of information system to automate educational process based on the Faculty of Informatics and Computer Science at NTUU "Igor Sikorsky Kyiv Polytechnic Institute". Existing educational systems of different higher education institutions had been studied; main realized functions of similar platforms were defined. As a result of research model, that enables insertion of students, teachers and other university personnel data, storing personal data and information about users' scientific works, and also is able to be integrated into existing university information space, has been obtained.

Keywords: educational process, information system, automatization, practical use.

1. Introduction

Information system of educational process automatization allows to solve one key issue of informational technologies integration. We are suggesting simple and effective approach to building powerful educational process automatization system, and also demonstrating practical usage aspect in different educational process – that is the goal of this research paper.

Product examination includes forming base system structure, introducing new information solutions, that will allow maximum automatization of documents insertion process performed by teachers, will include effective feedback mechanism. will enable platform integration into modern social networks and the Faculty of Informatics and Computer Science at NTUU "Igor Sikorsky Kyiv Polytechnic Institute" information space – that lets simple and accessible connection between system users.

2. References analysis and work perspective

It is hard to integrate modern technologies into different life aspects in the world full of developed computer systems. In the area of education, it is very important to create automated user control system with possibility of effective usage in educational, documentation preparations processes [10]. Informational technologies usage allows creation such a powerful system, that greatly simplifies mechanism of educational process management in educational institutions [6]. Apart from that, mass usage of similar systems enables more effective educational process of modern youth, that is greatly digitalized, and also simplifies access to scientific works of many authors, which might unite education and tuition into a single management process [16]. This might become a revolution in modern education perception [12].

Let's examine some examples of information educational management systems from different universities around the globe.

Work [18] presents multiple educational platforms of Russian educational institutions. Some examples of these systems are "Educational process quality management. Performance and attendance accounting" of Samara State Technical University, "Electronic university" of Kazan Federal University etc. Each of these systems realized only a part of base functionality, required for university work organization within separate information space. Most notable among them are platform for accounting students' attendance and performance, subjects lists and corresponding curriculums formation, automatic time schedule generation based on tutors' load [4, 7, 8].

Among other examples of introducing information educational management systems are multiple platforms developed in USA, e.g., information system from University of Colorado Boulder named "MyCUinfo". In comparison to Russian information systems, such platforms have very similar students' educational process management, but also has a list of additional functions that includes finances management, personal library access etc [15].

3. Research goals and tasks

This paper has a goal of information system to provide faculty with means of educational process automation architecture analysis and building. This also includes analysis and evaluation model of current provisions on educational process in at NTUU "Igor Sikorsky Kyiv Polytechnic Institute" for using it as a base for architecture creation.

Such a model requires according realization approach within some information system, that will be available for each educational process party: students, tutors, educational institution structural units etc. Another important factor is which software tools allow building a platform based on own model. Examination of such structural components is a task of this paper.

4. Research data and methods

Building effective educational process automatization involves solving multiple important problems. First of all, system should use modern information technology software tools to enable fast system access for each user, and also will effectively solve simultaneous access of significant users count. Secondly, this system should realize traditional information system creation model, which will serve as a base for different components to function properly. Thirdly, system should follow existing principles of educational process organization within specific educational institution.

Realization of this information system should take into consideration scalability issue, as users and database records count growth might lead to general system performance regression. Among other existing methods of computer systems scaling, the most suitable for current approach is horizontal scaling. During building distributed model system, which might be easily parallelized between different hardware units using containerization technologies and approach of creation modules as different services, usage of such a technique will allow ease of scalability to support growing users and requests count [2].

Accordingly, the central system component is relational database, which enables effective data storage in tables, its view and modification in different parts of system through internal database connections. PostgreSQL was the choice of a specific SQL-like database management system, which is fast and powerful open-source solution with possibility to scale up or out as a result of an increase in users or requests count. Main system software component, that also serves as connection to database, is built using modern version of PHP programming language with application of Symfony framework, that uses MVC pattern while preserving SOLID principles during development cycle. Frontend interface of platform will be written in JavaScript, which is the most popular browser programming language, with framework React, that has lost of pre-build forms and components to use.

Apart from listed components realization, this platform will allow getting feedback between systems users. For this purpose, the most popular social networks (e.g., Telegram, Facebook, Linkedin etc) will be integrated. Also, it is important to note, that platform should have relations with other university websites and information systems, and ideally these relations would be bidirectional. Thus, such a product will allow new program complex integration into a indivisible information space of faculty or even university in general.

As a result, system model is presented on fig. 1.

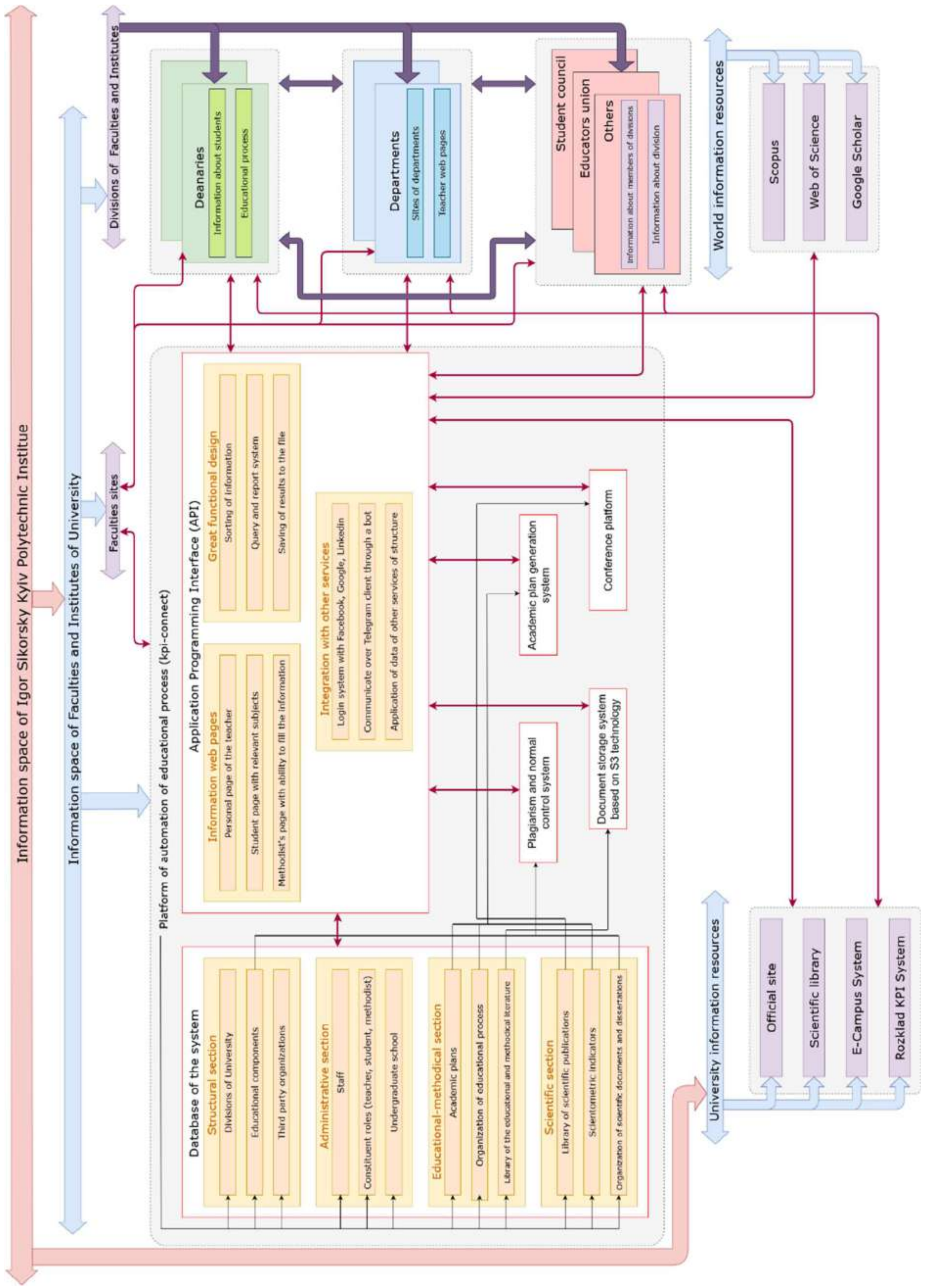

Information space of Igor Sikorsky Kyiv Polytechnic Institute
Information space of Faculties and Institutes of University
Divisions of Faculties and Institutes
Faculties sites
Deanaries
Information about students
Educational process
Departments
Sites of departments
Teacher web pages
Student council
Educators union
Others
Information about members of divisions
Information about division
World information resources
Scopus
Web of Science
Google Scholar
Platform of automation of educational process (kpi-connect)
Application Programming Interface (API)
Information web pages
Personal page of the teacher
Student page with relevant subjects
Methodist's page with ability to fill the information
Great functional design
Sorting of information
Query and report system
Saving of results to the file
Integration with other services
Login system with Facebook, Google, Linkedin
Communicate over Telegram client through a bot
Application of data of other services of structure
Plagiarism and normal control system
Document storage system based on S3 technology
Academic plan generation system
Conference platform
Database of the system
Structural section
Divisions of University
Educational components
Third party organizations
Administrative section
Staff
Constituent roles (teacher, student, methodist)
Undergraduate school
Educational-methodical section
Academic plans
Organization of educational process
Library of the educational and methodical literature
Scientific section
Library of scientific publications
Scientometric indicators
Organization of scientific documents and dissertations
University information resources
Official site
Scientific library
E-Campus System
Rozklad KPI System

Fig. 1. High-level system structure and relations with university information space

5. Research results

Applying platform description from fig. 1 development of individual system components was performed in accordance with general structure. Also, as a result of research, it is possible to define practical usage of each separate component, that is a part of global platform.

According to platform description, four main components may be highlighted, that are main parts of system. However, they are not the only ones allowed as system anticipates usage of additional components that distinguish it among other similar products.

Connection between database and system users is performed through API (Application Programming Interface). Its goal is to provide convenient access to information, that is stored in database, and also to allow effective and simple management of it and additional system capabilities.

According to description, let’s define system sections as follows.

5.1. Structural section

Structural section of platform is responsible for building educational structure within university. Presented entities enable university components hierarchy creation in accordance to principles of educational process at NTUU “Igor Sikorsky Kyiv Polytechnic Institute” [17]. It is possible to design and realize a base for university educational components using given entities.

Structural section organization model is illustrated on fig. 2.

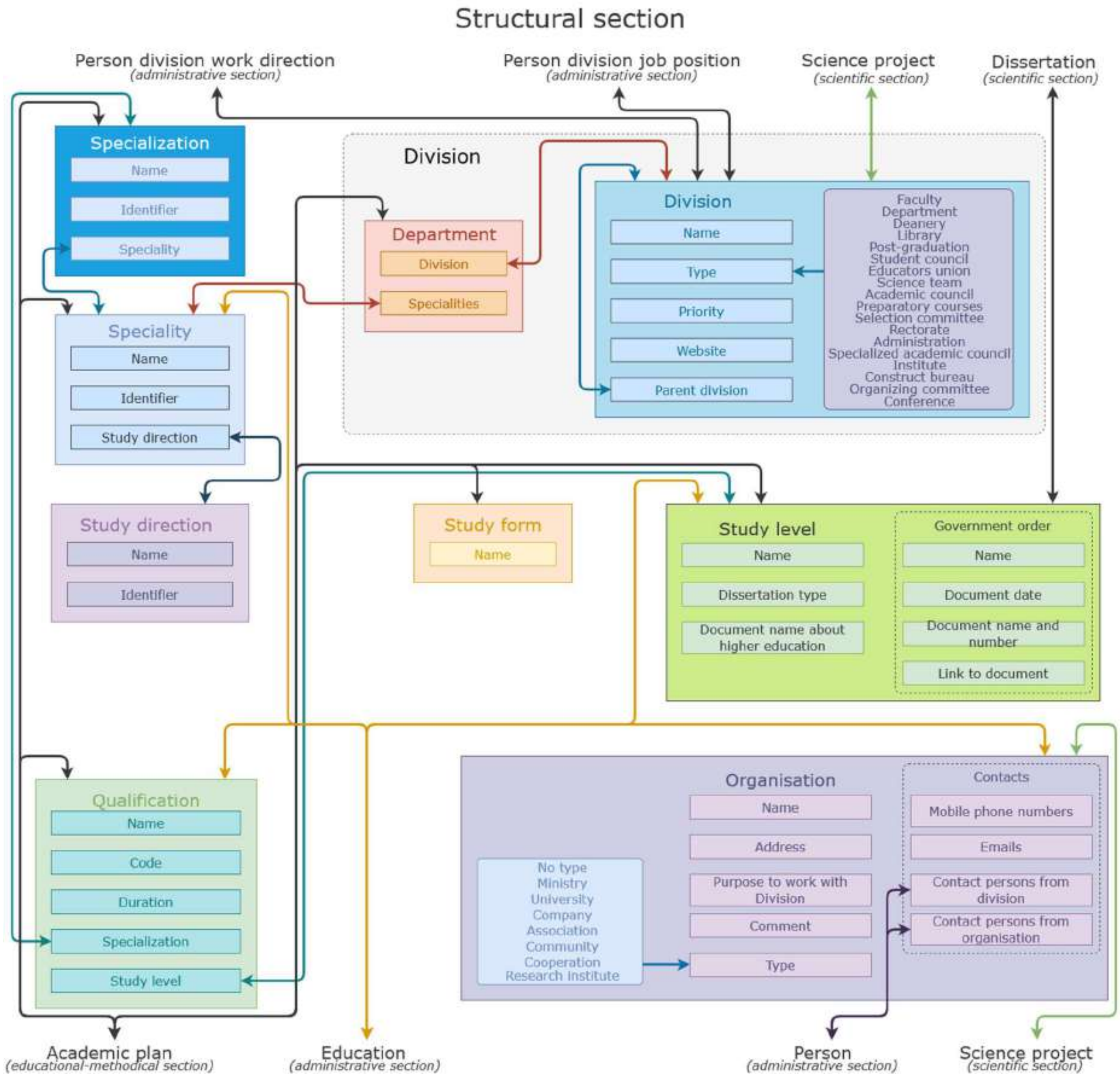


Fig. 2. Structural section of platform

5.2. Administrative section

Administrative section of platform forms personnel structure, which is base unit working with platform. This section describes base entity of Person, and key roles of system users – Student and Tutor. Interaction of Person and different parts of platform is defined within connections of current section entities.

Thus, we're having administrative section organization model on fig. 3.

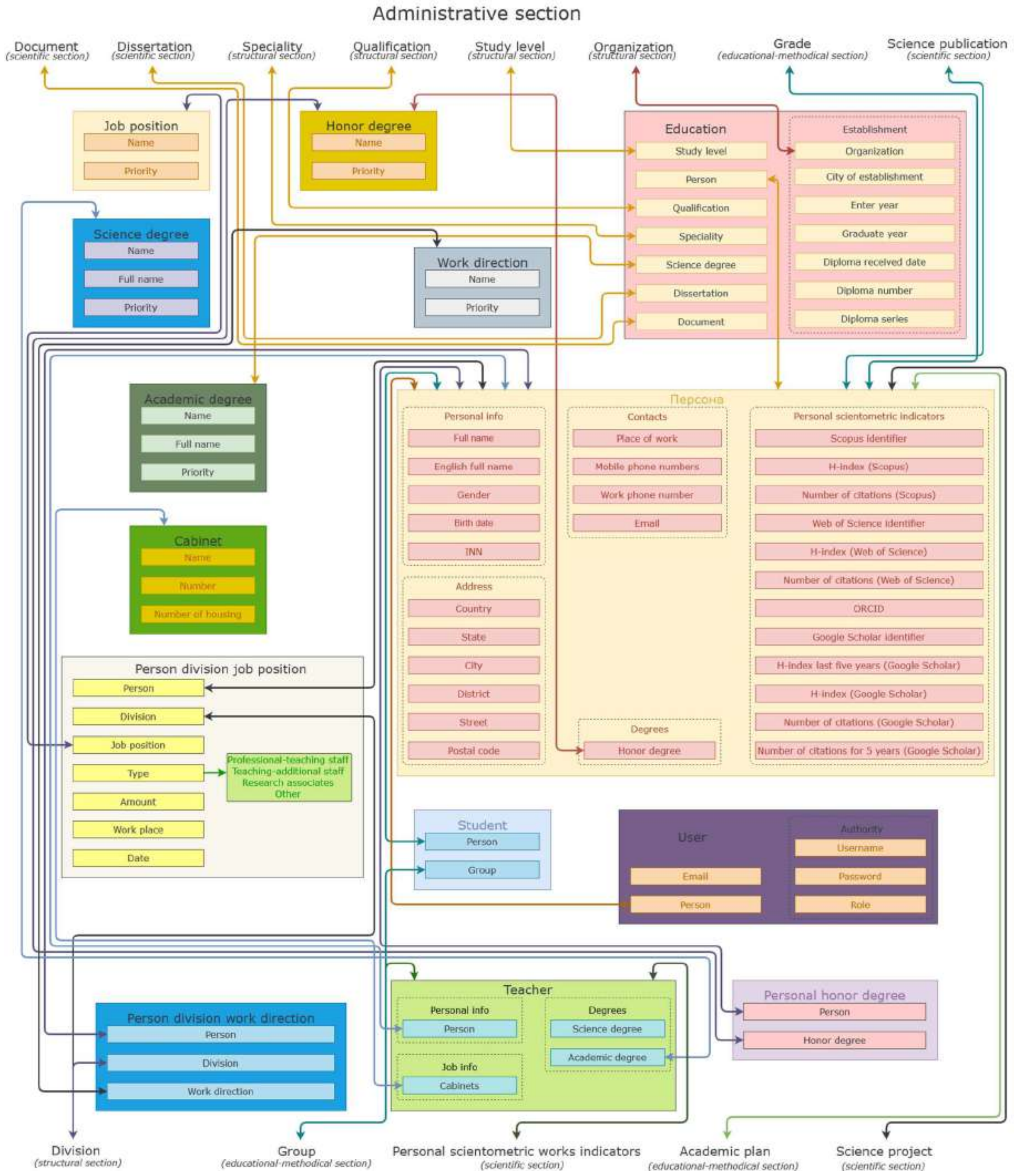


Fig. 3. Administrative section of system

5.3. Educational-methodical section

This section of system is responsible mainly for curriculum realization. Main component of this section is Curriculum (presented as Academic Plan), which is a base for planning subjects both for

Students and for Tutors. Another component is Group, which unites Students, Subjects and Grades through Curriculum.

Educational-methodical section organization model is presented on fig. 4.

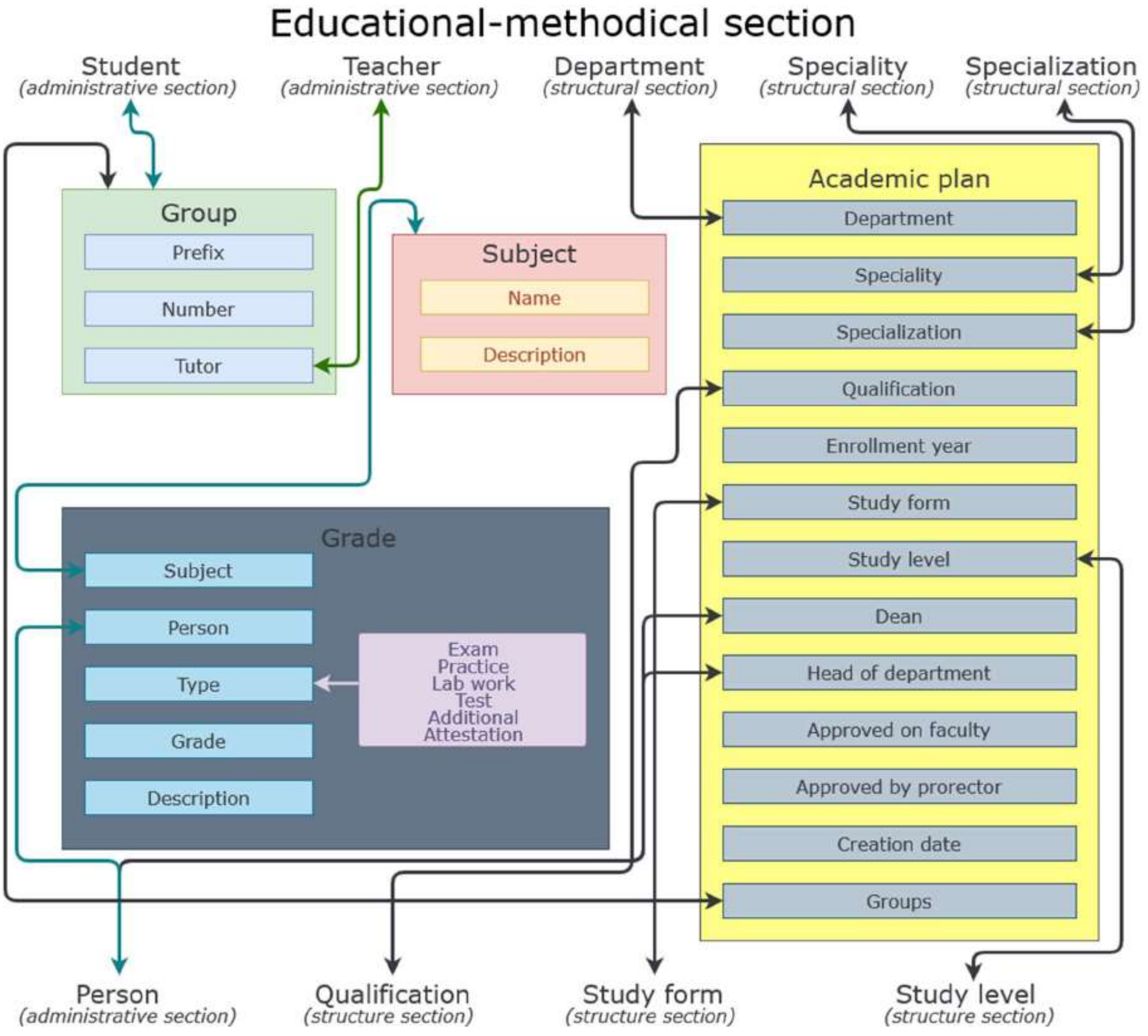

Educational-methodical section
Student (administrative section)
Teacher (administrative section)
Department (structural section)
Speciality (structural section)
Specialization (structural section)
Group
Prefix
Number
Tutor
Subject
Name
Description
Academic plan
Department
Speciality
Specialization
Qualification
Enrollment year
Study form
Study level
Dean
Head of department
Approved on faculty
Approved by prorector
Creation date
Groups
Grade
Subject
Person
Type
Grade
Description
Exam
Practice
Lab work
Test
Additional
Attestation
Person (administrative section)
Qualification (structure section)
Study form (structure section)
Study level (structure section)

Fig. 4. Educational-methodical section of system

5.4. Scientific section

Scientific section is responsible for scientific publications structure, and also for storing students' dissertations. Important component here is object storage, which performs uploading and storing documents inside of the system. Due to the nature of these files, it is unadvisable to keep them in traditional databases (both relational and NoSQL types) as it is relatively hard to effectively store and retrieve them, while their usually bigger size would mean additional load to the main component of the whole system. Apart from documents, other entities in this section include information of scientometric indicators of university tutors with possibility to extend them.

Scientific section organization model is presented on fig. 5.

6. Conclusions

Using faculty educational process automatization information system "KPI-Connect" has a number of advantages. First of all, described system implies dependent sequential structure Study direction – Specialty – Specialization and practically connected Qualification. Such an organization allows development of separate entity of Curriculum/Academic plan with possibility to develop automatic generation algorithm in future. Secondly, each key system component provides a possibility to create and generated highly convoluted entities based on universal entity of Person, then assign specific role to it, which will regulate available and visible information from user perspective. In virtue of this personnel organization, it is possible to uniquely and independently control each Person's educational process. Thirdly, research involves designing and forming unified compound structure of Curriculum (Academic plan) – it combines Student, Tutor, Subject and Grade through single data entity, that creates fast and flexible connections between Tutor and Student: the first one is able to conveniently store big amounts of educational process information, the latter is able to retrieve only relevant parts of it.

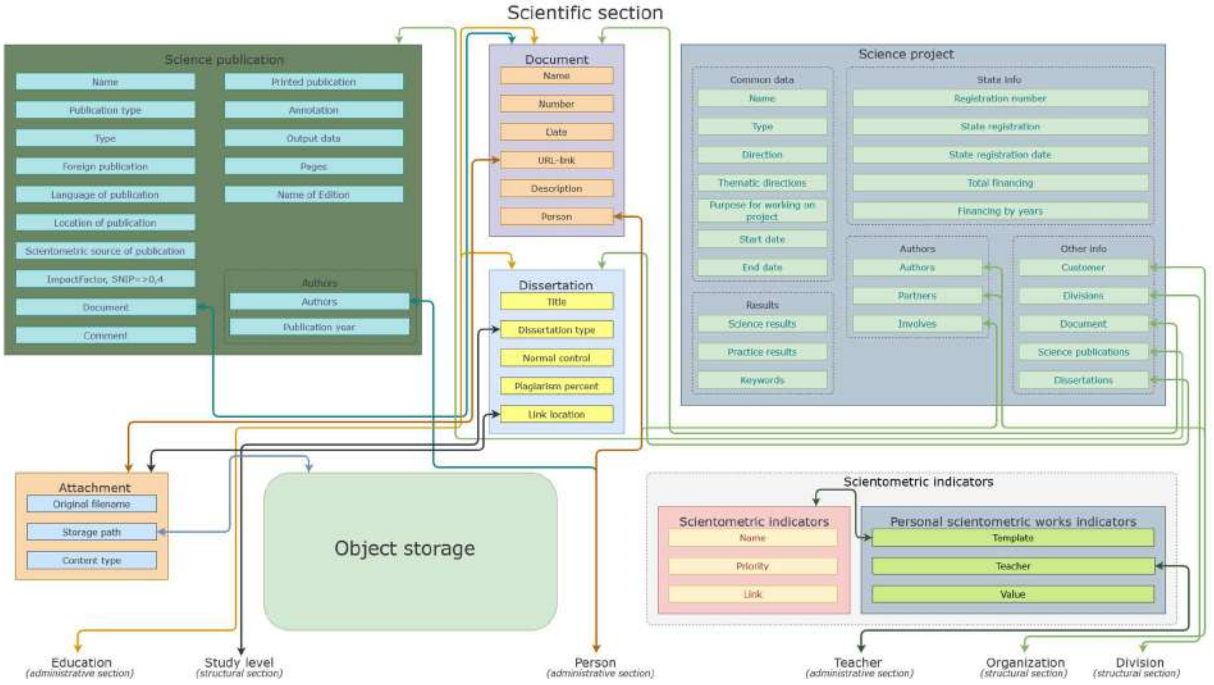


Fig. 5. Scientific section of system

Apart from that, obtained structure fully complies with principles of educational process organization at NTUU "Igor Sikorsky Kyiv Polytechnic Institute". As a result of that, such system can complement educational process and help with its management. And finally, object storage organization enables means to automate process of checking dissertations and scientific papers for plagiarism and compliance with formal requirements with appropriate systems.

References

[1] "Amazon Simple Storage Service User Guide," Amazon Web Services, Inc, 2021. Available: https://docs.aws.amazon.com/AmazonS3/latest/userguide/s3-userguide.pdf

[2] E. Banks, "What does 'scale out' vs. 'scale up' mean? - Packet Pushers," *Packet Pushers Interactive LLC*, 2017. https://packetpushers.net/scale-up-vs-scale-out/

[3] A. B. Bondi, "Characteristics of scalability and their impact on performance," in *Proceedings of the 2nd international workshop on Software and performance*, Ottawa, Ontario, Canada: Association for Computing Machinery, 2000, pp. 195–203. doi: https://doi.org/10.1145/350391.350432.

[4] M. Evdokimov and A. Mishhenko, "Information system for operational monitoring of students' progress and attendance. Development and implementation experience," *Vestnik Samarskogo gosudarstvennogo tehnicheskogo universiteta. Serija: Psihologo-pedagogicheskie nauki*, pp. 49–54, 2010, Available: https://cyberleninka.ru/article/n/informatsionnaya-sistema-operativnogo-kontrolya-uspevaemosti-i-poseschaemosti-studentov-opyt-razrabotki-i-vnedreniya/viewer

[5] M. Gehlawat, "School Management Information System: An Effective Tool for Augumenting the School Practices," *New Frontiers in Education: International Journal of Education & Research*, vol. 47, pp. 57–64, Jun. 2014.

[6] V. S. Gorjunov, “Information Systems in Education,” *Molodoj uchenyj*, vol. 5, no. 16, pp. 159–161, 2010, Available: https://moluch.ru/archive/16/1540/

[7] V. M. Gostev and R. H. Latypov, “Educational information environment of the Faculty of CMC of Kazan Federal University: the experience of formation and development,” *Sovremennye informacionnye tehnologii i IT-obrazovanie*, pp. 220–225, 2010, Available: https://cyberleninka.ru/article/n/obrazovatelnaya-informatsionnaya-sreda-fakulteta-vmk-kazanskogo-federalnogo-universiteta-opyt-formirovaniya-i-razvitiya/viewer

[8] T. N. Gur’janova T. N. and L. K. Karimova, “The use of information systems in the educational, scientific and administrative activities of the university (on the example of Kazan Federal University),” *Vestnik Kazanskogo tehnologicheskogo universiteta*, pp. 381–383, 2014, Available: https://cyberleninka.ru/article/n/primenenie-informatsionnyh-sistem-v-obrazovatelnoy-nauchnoy-i-adminictrativnoy-deyatelnosti-vuza-na-primere-kfu/viewer

[9] R. S. Hataeva, “Structural components of an automated management system of a modern innovative university,” *Vestnik universiteta*, no. 5, 2015, Available: https://cyberleninka.ru/article/n/strukturnye-komponenty-avtomatizirovannoy-sistemy-upravleniya-sovremennogo-innovatsionnogo-vuza/viewer

[10] N. Levenson and U. Boser, “The Promise of Education Information Systems. How Technology Can Improve School Management and Success,” *Center for American Progress*, p. 25, 2014, Available: https://files.eric.ed.gov/fulltext/ED561090.pdf

[11] J. Levy, “Management Information Systems in Education,” *Iris Software Group Ltd*, 2020. https://www.iris.co.uk/blog/management-information-systems-in-education/

[12] S. Montoya, “Why We Need Effective Education Management Information Systems | UNESCO UIS,” *UNESCO Institute of Statistics*, 2018. http://uis.unesco.org/en/blog/why-we-need-effective-education-management-information-systems

[13] K. S. S. Musti, “Management Information Systems for Higher Education Institutions,” in *Quality Management Implementation in Higher Education*, Hershey, PA: IGI Global, 2020, pp. 110–131. doi: https://doi.org/10.4018/978-1-5225-9829-9.ch006.

[14] “MyClarion Student Information System,” *Clarion University*. https://www.clarion.edu/about-clarion/computing-services/myclarion/index.html (accessed Nov. 19, 2023).

[15] “myCUinfo Available Features,” *University of Colorado Boulder*. https://spot.colorado.edu/~mycuinfo/help/features.html (accessed Nov. 19, 2023).

[16] S. Nussbaum-Beach, “A Futuristic Vision for 21st Century Education,” *ASCD Express*, vol. 6, no. 11, 2011, Available: http://web.archive.org/web/20201105231857/http://www.ascd.org/ascd-express/vol6/611-nussbaum-beach.aspx

[17] NTUU KPI im. Ighorja Sikorsjkogho, “Regulations on the Organization of the Educational Process in Igor Sigorsky Kyiv Polytechnic Institute,” 2020. https://document.kpi.ua/files/2020_7-124.pdf

[18] V. V. Senkin, “Possibilities of information systems in education management,” *Vestnik Juzhno-Ural’skogo gosudarstvennogo universiteta. Serija: Obrazovanie. Pedagogicheskie nauki*, vol. 41, no. 18, pp. 42–45, 2012, Available: https://cyberleninka.ru/article/n/vozmozhnosti-informatsionnyh-sistem-v-upravlenii-obrazovaniem/viewer

UDK 004.383
DOI: https://doi.org/10.20535/2708-4930.2.2021.251693

M. SERGIYENKO
O. MOLCHANOV
M. ORLOVA

MICROCONTROLLER FOR THE LOGIC TASKS

A new SM16 microcontroller architecture is proposed which is intended for the logic-intensive applications in the field-programmable gate array (FPGA). The microcontroller has the stack architecture which provides the implementation of the most of instructions for a single clock cycle. The short but fast programs are derived due to the 16-bit instructions, which code up to three independent operations, and intensive use of the threaded code style. The framework is developed which compiles the program, simulates it, and translates to the ROM. The developed SM16 core with additional three-stack blocks, hash-table, and instructions that accelerate the execution of parsing operations is used for efficient XML-document processing and can be frequently reconfigured to the given document grammar set. The parsing speed equals to one byte per 24 clock cycles.

Keywords: VHDL, XML, parser, FPGA, stack processor, grammar, FSM

1. Introduction

The evolution of the central processing unit (CPU) microarchitectures during decades was intended for increasing the speed of the usual computations in different fields. For this purpose, the instruction level parallelism is exploited in the directions of pipelining, superscalar computations, data and instruction caching, branch prediction, dynamic scheduling, speculative calculations, etc. As a result, a single processor could perform averagely up to two or more instructions per clock cycle with the frequency of several gigahertzes. These achievements are got at the costs of increasing the hardware volume by several decimal orders of magnitude and the power consumption up to dozens of Watts. But at present, the processor improvements stopped, in general, due to the Moore’s law and the Dennard scaling law limitations [1].

The next microarchitecture evolution is expected in the form of the architecture improvements in the application specific fields. However, the RISC architecture will be likely prevalent one. One of the successful approaches is based on the complex application-specific instructions implemented in the field programmable gate array (FPGA) which stays near CPU [2].

The logic decision-intensive algorithms are implemented in the modern microprocessors ineffectively. This is due to the fact of the frequent pipeline stalls when the branches are mispredicted [1]. One of the solutions to this problem is to go back to the non-pipelined CPUs. When CPU has the application-specific instruction set, it can have the minimized hardware volume. Hence, it has the minimized clock period, and could implement a single instruction for a single clock cycle including the logic branch instructions, and doing without pipelining. Such a CPU is considered in this work.

2. XML-Document Parsing. A Case Study

The Web service is based on the queries which are specific XML-documents. The essence of the XML-filtering is to detect the XML-queries which satisfy the given set of grammars, which number can achieve more than thousands. Parsing XML-queries results in a significant slowdown in the Web service performance [3]. The experience of using XML in the databases shows that XML-parsing is a major bottleneck in the productivity gains and can increase the transaction costs up to ten times and more [4].

The grammar of a particular query type is expressed using XML-query languages such as XPath [5]. In general, such a grammar is presented as a tuple $G = (N, T, S, P)$. Here, N is a finite set of the non-terminal symbols, T is a finite set of the terminal symbols, $S \subseteq N$ is a set of initial symbols, P is a set of rules in the form X → *ar*, where $X \in N$, $a \in T$, and *r* is a regular expression over *N*. The rule says that *X* originates the sub-trees with the root *a,* and children that satisfy the expression *r* [6].

A simple XML-parsing algorithm that validates the document in terms of a given regular tree grammar is described in [6]. The algorithm is implemented in a stack finite state machine (FSM), which has three stacks: P, Y, and S. Stack P stores the symbol sets from *N*. Stack Y stores the sets of rules from *P*. Stack S stores the lists of symbol sets from *N*. The algorithm traverses the document tree in-depth and triggers the event phrases of a document, which include the opening and closing tags.

So, we can see that the XML-parsing algorithm is a set of FSMs, each of them represents a single grammar. Such an algorithm must contain a lot of logic and comparing operations, and is supported by such data structure like a stack.

The XML text filtering is a difficult problem because it has to support the real time processing of wide streams of various XML-requests. Different methods and accelerators have been proposed to improve this task. There are software accelerators, like an XML-filter (XFilter), which are built as FSM implemented in software [7]. According to the LazyDFA method, weakly deterministic FSM is dynamically constructed for the XML filtering [8].

FPGA is an efficient solution for the hardware filtering of XML-queries. FSM that is constructed for a specific set of grammars is implemented in FPGA in [9]. The systems based on stack FSM, which are compiled from the given grammars, are shown in [10–12]. But each exchange of the grammar set affords the redesign of the whole project which lasts a lot of time.

An FSM skeleton is proposed in [13], which is capable of being reconfigured quickly without the FPGA project redesigning. This FSM skeleton becomes a working FSM after loading the transition conditions corresponding to a specific set of XML-requests. The disadvantages of these approaches consist of the high hardware redundancy and limitations of the processed document class.

Comparing the mentioned approaches, the following conclusions are done. The hardware systems have the highest performance, but they are designed for a limited number of XML-grammars and their reconfiguration is long-lasting. The reconfigurable FSM-based hardware filtering systems have excessive hardware costs and focus on a particular class of grammars. More flexible architectures that can provide both the high throughput and the ability to quickly be reconfigured to the arbitrary XML-grammar are required. And such architecture can be based on the microcontroller adapted to the logic tasks.

3. Stack Processors for the Logic Algorithm Programming

The conclusions of the previous chapters show that the new processor architecture for the implementation of the logic decision-intensive algorithms is of demand. Such architecture has to be capable to implement effectively FSMs which perform large algorithm sets.

Usually, the microcontrollers do this task very well. In [14] the experience of FSM programming in the ARM Cortex microcontroller architecture is shown. It is proven in it that the best results are achieved in this RISC architecture when it has the minimized number of pipelined stages, which is equal to two. By this condition, the delay of the logic branch is minimized up to two clock cycles. Note that the number of stages in the RISC processors usually varies from three to five and more.

The processors which are implemented in FPGA must be adapted to its properties. These properties are the implementation of the logic functions in the look-up tables (LUTs), which input number varies from four to eight and more, a sufficient number of available pipelining registers, RAM blocks with the latent delay of two clock cycles. Besides, one has to take into account that the wire delays in FPGA achieve the value of 40% – 90% of the critical path delay.

These factors decrease the working frequency of the FPGA processors in 3 – 10 times comparing to the ASIC implementation of the same architecture. For example, the clone MIPSfpga of the popular RISC architecture has the maximum clock frequency of 60 MHz, and the microcontroller PIC32MZ with the same architecture has 200 MHz [15]. Note, that the processor with the similar architecture implemented in the advanced IC technology achieves the clock frequency up to one or more gigahertz. The RISC microprocessor cores of the same bit width which are adapted to the FPGA architecture like Xilinx Microblaze, Altera Nios, have much higher maximum clock frequency which achieves the value of 300 MHz [16].

For the implementation of application-specific systems in FPGA, it is important to get the configurable microcontrollers that have both minimized hardware volume and minimized length of its firmware because the amount of the embedded RAM blocks have significantly limited volume.

The stack processor architecture is distinguished among all microprocessor architectures. In this architecture, the registered RAM is substituted to the stack of registers, which communicates both with ALU and the return stack. The essence of this architecture consists of the implicit addressing of the working registers, direct implementation of the algorithms in Polish postfix notation, wide use of very quick procedure calls. As a result, the instructions of this architecture have a short length and can be implemented in a single clock cycle. Since these instructions support algorithms that actively use the stack addressing and subroutines, the programs that are composed for this processor occupy very small memory volume [17].

Many authors have developed several projects of stack processors, which are implemented in FPGA and are available for reproduction [18 – 20]. All of them have 16-bit instructions and process 16-bit data. It is shown in [20], that the stack processor has approximately 2.5 times less program length than the program for the Xilinx MicroBlaze processor in the logic branch-intensive computations. In addition, all stack processors allow the designer to increase the instruction set. In this case, the appropriate changes should be made to the description of the processor at the register transfer level.

Consequently, the architecture of the stack processor provides both firmware amount and hardware costs minimization. In addition, it is easy to develop compilers for such architecture, because, as a rule, its instruction set is a subset of the Forth language operators. It is known that this language is convenient both for grammatical parsing of lines and for the interpretation of high-level language operators. The stack processor assembly language has the same syntax as the Forth language [17]. Therefore, it is attractive to develop the stack processor architecture, which gives not only minimized hardware costs but also simplified implementation of user instructions, which are adapted to the logic computations.

4. SM16 Stack Processor Architecture

To solve the logic-intensive problems including one described in Chapter 2 the SM16 CPU architecture was developed. This 16-bit processor has a common dual-stack architecture [17], which structure is shown in Fig. 1. The eight-bit SM8 microcontroller described in [21] has been developed to implement the data communication protocols and it is a predecessor of SM16. Many instruction operations and other features were inherited from the SM8 architecture.

CPU includes a program counter (PC), Data RAM block, Program ROM block, an instruction register (IR), return address stack (Rstack), data stack (Dstack), ALU. The T, N registers are the top registers of the DStack. Register R is the top of the RStack, which also plays the role of a loop counter.

The program is loaded into the Program ROM during the FPGA configuration. It can be exchanged without reconfiguring FPGA using the memory programming tool. The dual-port Data RAM downloads the data to be processed from the external devices in the DMA mode. A and B are the index registers, and the peripheral register Ri serves for the interprocessor communications.

The HTable ROM stores the hash table for the transcoding the long tags found in the XML-documents into the numbers. The PStack, Ystack, and SStack stacks perform the same functions as the P, Y, and S stacks of the FSM described in Chapter 2.

All instructions have a 16-bit width. The instruction has one to three op-code fields F1, F2, F3 (see Fig. 2). The field F1 codes the call, jump, return operations, counter decrement, and jump if it is not equal to zero (DJNZ). The field F2 codes all ALU operations, and F3 does data read and store operations in different modes including the register addressing with the address in the A or B register with the post-increment. The variable-length field D stores a constant, or a jump address.

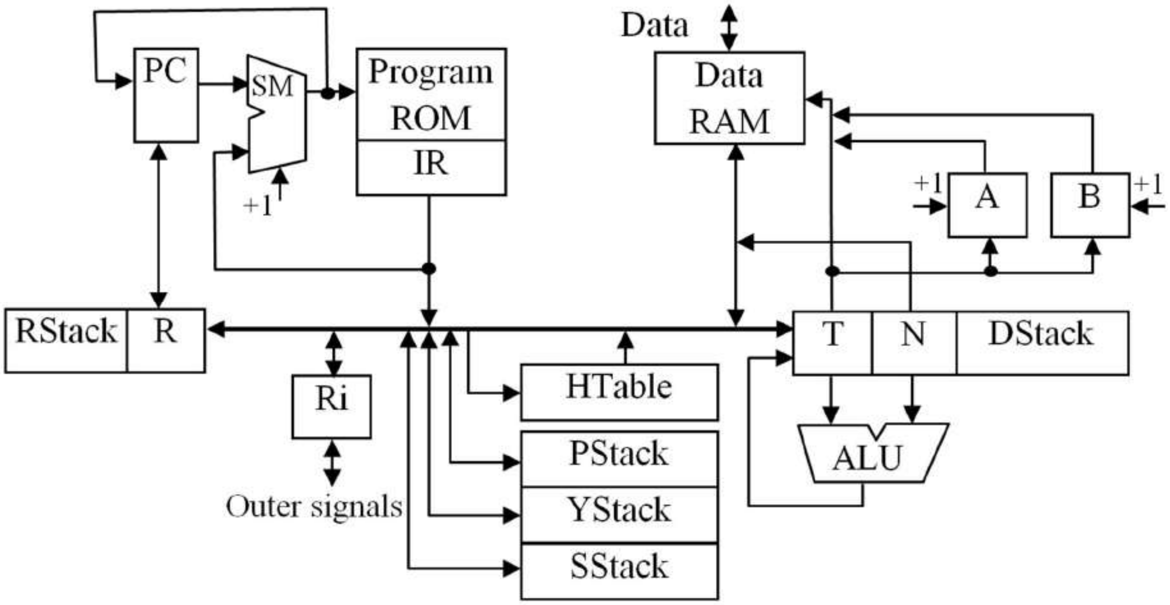


Fig. 1. SM16 processor structure

The instructions are executed in a single cycle except for the data read and long constant loading instructions that are executed in two cycles. This feature makes the architecture friendly to the algorithms that are branch intensive. The processor can perform up to three operations F1, F2, F3 in a single clock cycle. For example, two instructions

```
: L1      @B+
      !A+ DJNZ L1
```

perform a loop, which takes only 3 clock cycles, and in which an array addressed by the B register is moved to another memory place addressed by the A register. Here, according to the Forth syntaxis, “: L1” means the label, @B+, !A+ mean reading and writing operations, respectively, with the address post-increment.

The branch instruction lasts only a single clock cycle. This is achieved by the use of the ROM block output register as the instruction register IR and by feeding the next instruction address directly to the address input of this block bypassing the program counter PC (see Fig.1).

As a result, the logic branch-intensive algorithms are implemented without stalls. Consider an example of some FSM fragment coding which is shown in Fig.3. This example is shown in [14] as a result of the effective logic coding in the Cortex-M0+ microcontroller. Below, the code, proposed in [14] is compared with the similar code of SM16 processor.

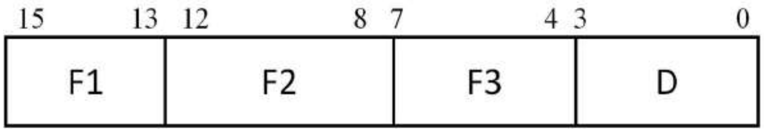


Fig. 2. SM16 instruction format

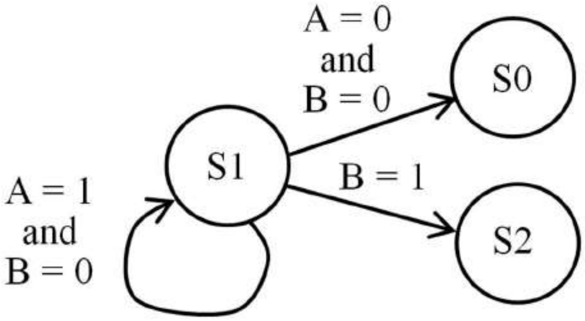


Fig. 3. Subgraph of some FSM

```
; Assembly code in ARM assembly syntax
S1:
    LDR       R0, [R4, #0x8]     ;     Read B
    CMP       R0, #1
    BEQ       S2                 ;     Goto S2 if B = 1
    LDR       R0, [R4, #0x4]     ;     Read A
    CMP       R0, #0
    BEQ       S0                 ;     Goto S0 if A = 0
    B         S1
```

Here, the code length is 14 bytes, the loop lasts 8 clock cycles taking into account that the branch operator takes 1 and 2 cycles for the undone and done branch respectively.

```
\ Assembly code in SM16 assembly syntax

          LIT 0          \ 0 to T for comparing
    : S1
          INR B          \ Read B
          LIT 1 XOR      \ comparing to 1
          IF  S2         \ Goto S2 if B = 1
          INR A          \ Read A
          IF  S1         \ Goto S0 if A = 0 else Goto S1
    : S0
```

Here, the code length is 10 bytes, the loop lasts 5 clock cycles. The instruction INR reads the respective peripheral register. We see that both the code length and the cycle duration are much less than in the effective example of the counterpart.

The stack processor architecture programming usually uses the threaded code style, i.e., the call instructions are placed very frequently. This instruction is implemented very quickly because the parameters are passed into the procedure in a natural fashion. So, the CALL and RET instructions are performed in the SR16 processor for a single clock cycle. The ability to insert a return operation in most instructions and combine it with a conditional branch reduces both the subroutine length and their duration. This helps both to speed-up the algorithm computing and to shorten the program dramatically. This makes it possible to obtain the programs of minimal length, which is important for the FPGA implementation. The next code shows an example of programming the deep if-then-else construction using these instructions.

```
: LONGIF
        TEST1
        LIT 1
        IF RET LIT 2
        TEST2
        IF RET LIT 3
;
```

This is the subroutine LONGIF which performs some logic testing, TEST1, TEST2 are calls of the subroutines which check some complex conditions, the character ‘;’ is the synonym of the RET instruction. As a result, each of the three testing outcomes returns figures 1, 2, or 3. This subroutine occupies only 12 bytes.

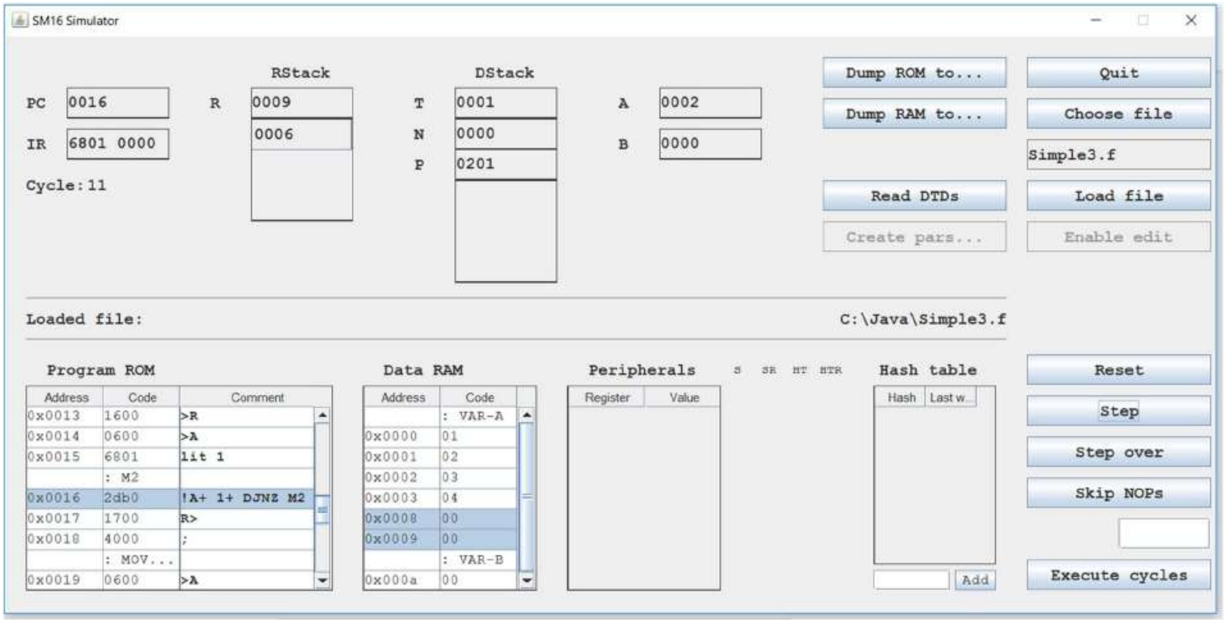
CPU has an interrupt system as well. Because the stack processor context has minimum volume, the interrupt overhead is also negligible. Due to large number of memory read instructions in the XML-parsing applications, the average run time of a single instruction is 1.2 clock cycles.

5. SM16 Processor Simulator

To develop the applications which perform the logic-intensive applications an SM16 processor simulator was designed. Its functions are: to compile the programs written in the SM16 assembly language, to load and simulate such a program, to inform about the syntax errors, to generate the VHDL files describing the Program ROM and Data RAM which contain the binary program and initial data codes, respectively.

This framework is able to read the document type definition (DTD) file which describes a set of XML-grammars and generates the SM16 program file to compute the respective parsing FSM. This program could both be modeled using the loaded XML-queries and be attached to the VHDL-model of the SM16 processor which is configured in FPGA.

The screenshot of the simulator frame is shown in Fig. 4.



PC 0016

IR 6801 0000

Cycle: 11

RStack

R 0009
0006

DStack

T 0001

N 0000

P 0201

A 0002

B 0000

Simple3.f

Loaded file: C:\Java\Simple3.f

Program ROM

Address	Code	Comment
0x0013	1600	>R
0x0014	0600	>A
0x0015	6801	lit 1
	: M2	
0x0016	2db0	!A+ 1+ DJNZ M2
0x0017	1700	R>
0x0018	4000	;
	: MOV...	
0x0019	0600	>A

Data RAM

Address	Code
	: VAR-A
0x0000	01
0x0001	02
0x0002	03
0x0003	04
0x0008	00
0x0009	00
	: VAR-B
0x000a	00

Register	Value

Hash	Last w...

Fig.4. Screenshot of the SM16 processor simulator

6. Experimental Results

The SM16 CPU is described in VHDL and is synthesized for configuring in FPGAs of different series. The results of configuring are shown in Table 1.

Table 1

SM16 Processor Core Parameters

FPGA series	LUTs, ALMs	Registers	Maximum clock frequency, MHz
Xilinx Spartan-6	721	116	102
Xilinx Artix-7	767	119	135
Xilinx Kintex-7	773	119	190
Intel Cyclone V	1001	1080	97

When configuring in Xilinx FPGA, all the stacks are mapped into LUTs effectively preserving low hardware volume and high speed. Much worse results are achieved in the Intel Cyclone chip because these stacks are implemented in the sets of registers.

Table 2 presents the results of the SM16 CPU configured in Xilinx Spartan-6 FPGA comparing to the other processor cores. The table analysis shows that the SM16 processor has higher performance as the b16-small [19] and J1 processor [20], which are the stack processors as well at the cost of higher hardware volume. It has a much higher speed than the well-known MSP430 processor [22] and somewhat loses to the Microblaze processor [23]. Nevertheless, it should be noted that the SM16 processor has a larger instruction set which is adapted to the logic algorithms especially to handle the XML-documents. Of course, it has much higher hardware volume than the 8-bit stack-based microcontroller SM8 [24].

Table 2

Comparing Different Processor Cores

Processor core	Bit-width	Hardware costs (LUTs)	Maximum clock frequency, MHz	Speed, MIPS
b16-small	16	280	100	50 MIPS
J1	16	342	106	70 MIPS
MSP430	16	1240	65	25 MIPS
Microblaze	32	2046	130	174 DMIPS
SM8	8	181	140	94 MIPS
SM16	16	721	116	96 MIPS

There are a few hardware implementations of XML parsers comparing to the software ones. Only hardware parsers XPA [25], SCBXP [26] provide both regular expression filtering and building the XML parsing tree.

A SM16 microcontroller was developed which is programed to implement the same tasks that these hardware parsers do. For this purpose, the additional three stacks and hash table were added to the CPU core as well as the instructions which support the parsing process. Among them the instruction HASH performs the hash function calculating of the XML key words with the speed of one character per 3 clock cycles. So, the keywords are substituted to the indexes which are compared to ones stored in the precompiled hash table.

The given grammar set is loaded to the simulator framework which generates both the program ROM model and the hash table model described in VHDL. As a result, the SM16 microcontroller can compute the XML queries at the speed of approximately 7.5 megabytes per second. Table 3 shows the characteristics of the mentioned hardware XML parsers and the proposed one.

Thus, the SM16 parser has only one and a half worse the performance-hardware ratio than the XRA and SCBXP parsers. However, an SM16-based solution can simultaneously support almost any number of XML grammars. This qualitatively distinguishes the developed parsing method and SM16 microcontroller from other hardware solutions.

Table 3

Hardware Characteristics of different XML parsers

XML-parser	Hardware costs (LUTs)	Clock frequency, MHz	Throughput, MB/s
XPA	9200	125	125
SCBXP	29200	33	200 – 500
SM16	880	180	7,5

7. Conclusion

A new SM16 microcontroller architecture is proposed which is intended for the logic-intensive applications in FPGA which is based on the stack architecture. The short but fast programs are derived due to the 16-bit instructions, which code up to three independent operations, and intensive use of the threaded code style. The framework is developed which compiles the program, simulates it, and translates to the ROM. The developed SM16 core with additional three stacks, hash-table, and instructions that accelerate the execution of parsing operations is used for efficient XML-document processing and can be frequently reconfigured to the given document grammar set. This system is not only capable of processing the XML-documents efficiently but also can be quickly reconfigured to process the documents of other grammars.

References

[1] J. L. Hennessy, D. A. Patterson, and A. C. Arpaci-Dusseau, *Computer architecture: a quantitative approach*. Cambridge, Ma: Morgan Kaufmann, 2019.

[2] J. L. Hennessy and D. A. Patterson, “A new golden age for computer architecture,” *Communications of the ACM*, vol. 62, no. 2, pp. 48–60, Jan. 2019, doi: https://doi.org/10.1145/3282307.

[3] M. R. Head, M. Govindaraju, R. van Engelen, and W. Zhang, “Benchmarking XML Processors for Applications in Grid Web Services,” in *SC ’06: Proceedings of the 2006 ACM/IEEE Conference on Supercomputing*, pp. 30–30. doi: https://doi.org/10.1109/SC.2006.14.

[4] M. Nicola and J. John, “XML parsing,” in *Proc. of the 20th Int. Conf. on Information and Knowledge Management*, Nov. 2003, pp. 175–178. doi: https://doi.org/10.1145/956863.956898.

[5] “XML Path Language Version 1.0,” *W3C*, Mar. 21, 2017. http://www.w3.org/TR/xpath (accessed Nov. 11, 2019).

[6] M. Murata, D. Lee, M. Mani, and K. Kawaguchi, “Taxonomy of XML schema languages using formal language theory,” *ACM Transactions on Internet Technology (TOIT)*, vol. 5, no. 4, pp. 660–704, Nov. 2005, doi: https://doi.org/10.1145/1111627.1111631.

[7] M. Altinel and M. J. Franklin, “Efficient filtering of XML documents for selective dissemination of information,” in *Proc. of the 26-th International Conference on Very Large Data Bases*, Jan. 2000, pp. 53–64.

[8] T. Green, A. Gupta, G. Miklau, M. Onizuka, and D. Suciu, “Processing XML streams with deterministic automata and stream indexes,” *ACM Transactions on Database Systems*, vol. 29, no. 4, pp. 752–788, Dec. 2004, doi: https://doi.org/10.1145/1042046.1042051.

[9] J. V. Lunteren, T. Engbersen, J. Bostian, B. Carey, and C. Larsson, “XML accelerator engine,” in *1st International Workshop on High Performance XML Processing*, 2004, pp. 1–4.

[10] R. Mueller, J. Teubner, and G. Alonso, “Streams on wires,” *Proceedings of the VLDB Endowment*, vol. 2, no. 1, pp. 229–240, Aug. 2009, doi: https://doi.org/10.14778/1687627.1687654.

[11] R. Moussalli, M. Salloum, W. Najjar, and V. J. Tsotras, “Massively parallel XML twig filtering using dynamic programming on FPGAs,” in *2011 IEEE 27th International Conference on Data Engineering*, pp. 948–959. doi: https://doi.org/10.1109/ICDE.2011.5767899.

[12] A. Mitra, M. Vieira, P. Bakalov, V. Tsotras, and W. Najjar, “Boosting XML filtering through a scalable FPGA-based architecture.,” in *Proc. of the 4th Biennal Conference on Innovative Data Systems Research*, 2009, pp. 1–10.

[13] J. Teubner, L. Woods, and C. Nie, “XLynx — an FPGA-based XML filter for hybrid XQuery processing,” *ACM Transactions on Database Systems*, vol. 38, no. 4, pp. 1–39, Nov. 2013, doi: https://doi.org/10.1145/2536800.

[14] J. Yiu, “Software based Finite State Machine (FSM) with general purposeprocessors,” ARM, Jan. 2013.

[15] S. L. Harris *et al.*, “MIPSfpga: using a commercial MIPS soft-core in computer architecture education,” *IET Circuits, Devices & Systems*, vol. 11, no. 4, pp. 283–291, Apr. 2017, doi: https://doi.org/10.1049/iet-cds.2016.0383.

[16] J Nurmi, Ed., *Processor Design. System-on-Chip Computing for ASICs and FPGAs*. Springer Dordrecht, 2007. doi: https://doi.org/10.1007/978-1-4020-5530-0.

[17] P. Koopman, *Stack computers: the new wave*. CA: Ellis Horwood, Mountain View Press, 1989.

[18] W. Leong, P. K. Tsang, and T. K. Lee, “A FPGA based Forth microprocessor,” in *Proceedings. IEEE Symposium on FPGAs for Custom Computing Machines (Cat. No.98TB100251)*, pp. 254–255. doi: https://doi.org/10.1109/FPGA.1998.707903.

[19] B. Paysan, “b16 — small — Less is More,” in *Proc. EuroForth 2004*, Jul. 2006.

[20] J. Bowman and W. Garage, “J1: a small Forth CPU Core for FPGAs,” in *Proc. EuroForth’2010*, Jan. 2010.

[21] A. Sergiyenko, O. Molchanov, and M. Orlova, “NanoProcessor for the Small Tasks,” in *2019 IEEE 39th International Conference on Electronics and Nanotechnology (ELNANO)*, pp. 674–677. doi: https://doi.org/10.1109/ELNANO.2019.8783555.

[22] O. Girard, “Using the MicroBlaze Processor to Accelerate Cost-Sensitive Embedded System Development,” *OpenCores*, 2013. http://opencores.org (accessed Nov. 11, 2019).

[23] V. Kale, “Using the MicroBlaze Processor to Accelerate Cost-Sensitive Embedded System Development,” *Xilinx*, 2016. https://docs.xilinx.com/v/u/en-US/wp469-microblaze-for-cost-sensitive-apps (accessed Nov. 11, 2019).

[24] O. Molchanov, M. Orlova, and A. Sergiyenko, “Software/Hardware Codesign of the Microprocessor for the Serial Port Communications,” in *Advances in Computer Science for Engineering and Education II*, Z. Hu, S. Petoukhov, I. Dychka, and M. He, Eds., Cham: Springer International Publishing, 2020, pp. 238–246.

[25] Z. Dai, N. Ni, and J. Zhu, “A 1 cycle-per-byte XML parsing accelerator,” in *FPGA ’10: Proc. of the 18th ann. ACM/SIGDA int. symp. on Field programmable gate arrays.* , Pennsylvania State University, Feb. 2010. doi: https://doi.org/10.1145/1723112.1723148.

[26] F. ElHassan and D. Ionescu, “SCBXP: An Efficient CAMBased XML Parsing Technique in Hardware Environments,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 11, pp. 1879–1887, doi: https://doi.org/10.1109/TPDS.2011.51.

ABSTRACT

UDK: 004.77, 004.728
DOI: https://doi.org/10.20535/2708-4930.2.2021.244169

METHODOLOGY OF NETWORK ENVIRONMENT TESTING FOR IoT DEVICES

(p. 4 – 11)

V. HER
Department of Computer Engineering National Technical University of Ukraine
"Igor Sikorsky Kyiv Politechnic Institute" Kyiv, Ukraine, Ukraine
ORCID: https://orcid.org/0000-0001-9044-1499

V. TARANIUK
QA Department GlobalLogic Ukraine Kyiv, Ukraine, Ukraine
ORCID: https://orcid.org/0000-0001-9044-1499

V. TKACHENKO
Department of Computer Engineering National Technical University of Ukraine
"Igor Sikorsky Kyiv Politechnic Institute" Kyiv, Ukraine, Ukraine
ORCID: https://orcid.org/0000-0002-1080-5932

I. KLYMENKO
Department of Computer Engineering National Technical University of Ukraine
"Igor Sikorsky Kyiv Politechnic Institute" Kyiv, Ukraine, Ukraine
ORCID: https://orcid.org/0000-0001-5345-8806

S. NIKOLSKY
Department of Computer Engineering National Technical University of Ukraine
"Igor Sikorsky Kyiv Politechnic Institute" Kyiv, Ukraine, Ukraine
ORCID: https://orcid.org/0000-0003-4893-3339

The article reviews methods and technologies for testing the network environment of embedded systems and writing test documentation. As an example, a testing technique based on a defect report has been developed. A performance test was developed to check the load of the embedded device's network environment using special bash scripts for performance testing.

Keywords**:** IoT, embedded system, test case, defect report, troubleshooting, performance testing.

UDC 004.056.53
DOI: https://doi.org/10.20535/2708-4930.2.2021.244178

ONE APPROACH TO ACCELERATE THE EXPONENTIATION ON GALOIS FIELDS FOR DATA PROTECTION CRYPTOGRAPHIC SYSTEMS

(p. 12 – 18)

O. MARKOVSKYI
National Technical University of Ukraine
"Igor Sikorsky Kyiv Politechnic Institute" Kyiv, Ukraine
ORCID: https://orcid.org/0000-0003-3483-4233

O. RUSANOVA
National Technical University of Ukraine
"Igor Sikorsky Kyiv Politechnic Institute" Kyiv, Ukraine, Ukraine
ORCID: https://orcid.org/0000-0003-0145-3012

AL-MRAYT GHASSAN ABDEL JALIL HALIL
National Technical University of Ukraine
"Igor Sikorsky Kyiv Politechnic Institute" Kyiv, Ukraine, Ukraine
ORCID: https://orcid.org/0000-0002-1610-1119

O. KOT
National Technical University of Ukraine
“Igor Sikorsky Kyiv Politechnic Institute” Kyiv, Ukraine, Ukraine
ORCID: https://orcid.org/0000-0001-5498-4170

The new approach to accelerate the computational implementation of the basic for a wide range of cryptographic data protection mechanisms operation of exponentiation on Galois Fields have been proposed. The approach is based on the use of a specific property of a polynomial square and the Montgomery reduction. A new method of squaring reduces the amount of computation by 25% compared to the known ones. Based on the developed method, the exponentiation on Galois Fields procedure has been modified, which allows to reduce the amount of calculations by 20%.

***Keywords*:** multiplication operation on Galois fields, cryptographic algorithms based on Galois Fields algebra, Galois Fields exponentiation, Montgomery reduction.

UDC 004.383.3, 004.056, 004.8
DOI: https://doi.org /10.20535/2708-4930.2.2021.244186

OPTIMAL CONSTRUCTION OF THE PATTERN MATRIX FOR PROBABILISTIC NEURAL NETWORKS IN TECHNICAL DIAGNOSTICS BASED ON EXPERT ESTIMATIONS

(p. 19 – 25)

V. ROMANUKE
Polish Naval Academy: Gdynia, Poland, Poland
ORCID: https://orcid.org/0000-0003-3543-3087

In the field of technical diagnostics, many tasks are solved by using automated classification. For this, such classifiers like probabilistic neural networks fit best owing to their simplicity. To obtain a probabilistic neural network pattern matrix for technical diagnostics, expert estimations or measurements are commonly involved. The pattern matrix can be deduced straightforwardly by just averaging over those estimations. However, averages are not always the best way to process expert estimations. The goal is to suggest a method of optimally deducing the pattern matrix for technical diagnostics based on expert estimations. The main criterion of the optimality is maximization of the performance, in which the subcriterion of maximization of the operation speed is included. First of all, the maximal width of the pattern matrix is determined. The width does not exceed the number of experts. Then, for every state of an object, the expert estimations are clustered. The clustering can be done by using the *k*-means method or similar. The centroids of these clusters successively form the pattern matrix. The optimal number of clusters determines the probabilistic neural network optimality by its performance maximization. In general, most results of the error rate percentage of probabilistic neural networks appear to be near-exponentially decreasing as the number of clustered expert estimations is increased. Therefore, if the optimal number of clusters defines a too “wide” pattern matrix whose operation speed is intolerably slow, the performance maximization implies a tradeoff between the error rate percentage minimum and maximally tolerable slowness in the probabilistic neural network operation speed. The optimal number of clusters is found at an asymptotically minimal error rate percentage, or at an acceptable error rate percentage which corresponds to maximally tolerable slowness in operation speed. The optimality is practically referred to the simultaneous acceptability of error rate and operation speed.

Keywords: technical diagnostics, probabilistic neural network, pattern matrix, expert estimations, clustering, performance maximization.

UDC 004.855.5
DOI: https://doi.org/10.20535/2708-4930.2.2021.244188

VECTOR SPACE MODELS OF KYIV CITY PETITIONS

(p. 26 – 34)

R. SHAPTALA
Educational and Research Complex “Institute for Applied Systems Analysis”
National Technical University of Ukraine
“Igor Sikorsky Kyiv Politechnic Institute” Kyiv, Ukraine
ORCID: https://orcid.org/0000-0002-4367-5775

G. KYSELOV
Educational and Research Complex "Institute for Applied Systems Analysis"
National Technical University of Ukraine
"Igor Sikorsky Kyiv Politechnic Institute" Kyiv, Ukraine
ORCID: https://orcid.org/0000-0003-2682-3593

In this study, we explore and compare two ways of vector space model creation for Kyiv city petitions. Both models are built on top of word vectors based on the distributional hypothesis, namely Word2Vec and FastText. We train word vectors on the dataset of Kyiv city petitions, preprocess the documents, and apply averaging to create petition vectors. Visualizations of the vector spaces after dimensionality reduction via UMAP are demonstrated in an attempt to show their overall structure. We show that the resulting models can be used to effectively query semantically related petitions as well as search for clusters of related petitions. The advantages and disadvantages of both models are analyzed.

***Keywords*:** vector space model, FastText, Word2Vec, petitions analysis, UMAP.

UDC 004.383
DOI: https://doi.org/10.20535/2708-4930.2.2021.244191

LOCAL FEATURE EXTRACTION IN IMAGES

(p. 35 – 47)

P. SERHIIENKO
National Technical University of Ukraine
"Igor Sikorsky Kyiv Politechnic Institute" Kyiv, Ukraine
ORCID: https://orcid.org/0000-0003-3030-0074

A. SERGIYENKO
National Technical University of Ukraine
"Igor Sikorsky Kyiv Politechnic Institute" Kyiv, Ukraine
ORCID: https://orcid.org/0000-0001-5965-1789

M. ORLOVA
National Technical University of Ukraine
"Igor Sikorsky Kyiv Politechnic Institute" Kyiv, Ukraine
ORCID: https://orcid.org/0000-0002-6617-4631

The methods of the local feature point extraction are analyzed. The analysis shows that the most effective detectors are based on the brightness gradient determination. They usually use the Harris angle detector, which is complex in calculations. The algorithm complexity minimization contradicts both the detector effectiveness and to the high dynamic range of the analyzed image. As a result, the high-speed methods could not recognize the feature points in the heavy luminance conditions.

The modification of the high dynamic range (HDR) image compression algorithm based on the Retinex method is proposed. It contains an adaptive filter, which preserves the image edges. The filter is based on a set of feature detectors performing the Harris-Laplace transform which is much simpler than the Harris angle detector. A prototype of the HDR video camera is designed which provides sharp images. Its structure simplifies the design of the artificial intelligence engine, which is implemented in FPGA of medium or large size.

***Keywords*:** FPGA, feature extraction, HDR, pattern recognition, artificial intelligence.

UDC 004.383
DOI: https://doi.org/10.20535/2708-4930.2.2021.244189

GIF IMAGE HARDWARE COMPRESSORS

(p. 48 – 55)

I. MOZGHOVYI
National Technical University of Ukraine
"Igor Sikorsky Kyiv Politechnic Institute" Kyiv, Ukraine
ORCID: https://orcid.org/0000-0001-5469-486X

A. SERGIYENKO
National Technical University of Ukraine
“Igor Sikorsky Kyiv Politechnic Institute” Kyiv, Ukraine
ORCID: https://orcid.org/0000-0001-5965-1789

R. YERSHOV
Chernihiv National University of Technology: Chernihiv, Ukraine
ORCID: https://orcid.org/0000-0002-0267-2906

Increasing requirements for data transfer and storage is one of the crucial questions now. There are several ways of high-speed data transmission, but they meet limited requirements applied to their narrowly focused specific target. The data compression approach gives the solution to the problems of high-speed transfer and low-volume data storage. This paper is devoted to the compression of GIF images, using a modified LZW algorithm with a tree-based dictionary. It has led to a decrease in lookup time and an increase in the speed of data compression, and in turn, allows developing the method of constructing a hardware compression accelerator during the future research.

***Keywords*:** FPGA, GIF, lossless compression, image compression, dictionary, hardware acceleration

UDC 004.414.22
DOI: HTTPS://DOI.ORG/10.20535/2708-4930.2.2021.247770

ARCHITECTURAL REVIEW AND CONCEPTUAL DEVELOPMENT OF FACULTY INFORMATION SYSTEM “KPI-CONNECT”

(p. 56 – 63)

I. KLYMENKO
National Technical University of Ukraine
“Igor Sikorsky Kyiv Politechnic Institute” Kyiv, Ukraine
ORCID: https://orcid.org/0000-0001-5345-8806

Y. BUTSKYI
National Technical University of Ukraine
“Igor Sikorsky Kyiv Politechnic Institute” Kyiv, Ukraine
ORCID: https://orcid.org/0000-0003-1358-2583

K. HRYSHCHENKO
National Technical University of Ukraine
“Igor Sikorsky Kyiv Politechnic Institute” Kyiv, Ukraine

ORCID: https://orcid.org/0000-0002-8186-960X

M. SIVACHENKO
National Technical University of Ukraine
“Igor Sikorsky Kyiv Politechnic Institute” Kyiv, Ukraine
ORCID: https://orcid.org//0000-0001-8661-1563

V. KRYVETS
National Technical University of Ukraine
“Igor Sikorsky Kyiv Politechnic Institute” Kyiv, Ukraine
ORCID: https://orcid.org/0000-0002-6686-0092

D. KRYVOSHEI
National Technical University of Ukraine
“Igor Sikorsky Kyiv Politechnic Institute” Kyiv, Ukraine
ORCID: https://orcid.org/0000-0003-2236-3710

D. NGUEN
National Technical University of Ukraine
“Igor Sikorsky Kyiv Politechnic Institute” Kyiv, Ukraine
ORCID: https://orcid.org/0000-0001-8254-3540

This paper is dedicated to development model of information system to automate educational process based on the Faculty of Informatics and Computer Science at NTUU “Igor Sikorsky Kyiv Polytechnic Institute”. Existing educational systems of different higher education institutions had been studied; main realized functions of similar platforms were defined. As a result of research model, that enables insertion of students, teachers and other university personnel data, storing personal data and information about users’ scientific works, and also is able to be integrated into existing university information space, has been obtained.

Keywords: educational process, information system, automatization, practical use.

UDK 004.383
DOI: HTTPS://DOI.ORG/10.20535/2708-4930.2.2021.251693

MICROCONTROLLER FOR THE LOGIC TASKS

(p. 64 – 72)

A. SERGIYENKO
National Technical University of Ukraine
“Igor Sikorsky Kyiv Politechnic Institute” Kyiv, Ukraine
ORCID: http://orcid.org/0000-0001-5965-1789

O. MOLCHANOV
National Technical University of Ukraine
“Igor Sikorsky Kyiv Politechnic Institute” Kyiv, Ukraine
ORCID: https://orcid.org/0000-0001-8384-0918

M. ORLOVA
National Technical University of Ukraine
“Igor Sikorsky Kyiv Politechnic Institute” Kyiv, Ukraine
ORCID: https://orcid.org/0000-0002-6617-4631

A new SM16 microcontroller architecture is proposed which is intended for the logic-intensive applications in the field-programmable gate array (FPGA). The microcontroller has the stack architecture which provides the implementation of the most of instructions for a single clock cycle. The short but fast programs are derived due to the 16-bit instructions, which code up to three independent operations, and intensive use of the threaded code style. The framework is developed which compiles the program, simulates it, and translates to the ROM. The developed SM16 core with additional three-stack blocks, hash-table, and instructions that accelerate the execution of parsing operations is used for efficient XML-document processing and can be frequently reconfigured to the given document grammar set. The parsing speed equals to one byte per 24 clock cycles.

Keywords**:** VHDL, XML, parser, FPGA, stack processor, grammar, FSM

АНОТАЦІЇ

УДК: 004.77, 004.728
DOI: https://doi.org/10.20535/2708-4930.2.2021.244169

МЕТОДИКА ТЕСТУВАННЯ МЕРЕЖЕВОГО СЕРЕДОВИЩА ДЛЯ IoT ПРИСТРОЇВ

(стор. 4 – 11)

В. ГЕР
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: https://orcid.org/0000-0001-9044-1499

В. ТАРАНЮК
«GlobalLogic Ukraine», Київ Україна
ORCID: https://orcid.org/0000-0001-9044-1499

В. ТКАЧЕНКО
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: https://orcid.org/0000-0002-1080-5932

І. КЛИМЕНКО
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: https://orcid.org/0000-0001-5345-8806

С. НІКОЛЬСЬКИЙ
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: https://orcid.org/0000-0003-4893-3339

У статті оглянуто методи та технології тестування мережного оточення вбудованих систем і написання тестової документації. В якості прикладу розроблено техніка тестування на основі звіту про дефекти. Розроблено тест продуктивності для перевірки навантаження мережного оточення вбудованого пристрою з використанням спеціальних bash-скриптів для тестування продуктивності.

Ключові слова: IoT, вбудована система, тестовий приклад, звіт про дефекти, усунення несправностей, тестування продуктивності.

УДК 004.056.53
DOI: https://doi.org/10.20535/2708-4930.2.2021.244178

ОДИН ПІДХІД ДО ПРИСКОРЕННЯ ПОКАЗУВАННЯ НА ПОЛЯХ ГАЛУА ДЛЯ КРИПТОГРАФІЧНИХ СИСТЕМ ЗАХИСТУ ДАНИХ

(стор. 12 – 18)

О. МАРКОВСЬКИЙ
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: https://orcid.org/0000-0003-3483-4233

О. РУСАНОВА
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: https://orcid.org/0000-0003-0145-3012

Г. А. ДЖ. АЛЬ-МРАЯТ
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: https://orcid.org/0000-0002-1610-1119

О. КОТ
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: https://orcid.org/0000-0001-5498-4170

Запропоновано новий підхід до прискорення обчислювальної реалізації базової для широкого кола механізмів криптографічного захисту даних операції піднесення до степеня за полями Галуа. Підхід заснований на використанні специфічної властивості квадрата полінома та редукції Монтгомері. Новий метод зведення в квадрат скорочує обсяг обчислень на 25% порівняно з відомими. На основі розробленого методу модифіковано процедуру піднесення до степеня на полях Галуа, що дозволяє зменшити обсяг обчислень на 20%.

Ключові слова: операція множення на поля Галуа, криптографічні алгоритми на основі алгебри полів Галуа, піднесення до степеня за полями Галуа, редукція Монтгомері.

УДК 004.383.3, 004.056, 004.8
DOI: https://doi.org/10.20535/2708-4930.2.2021.244186

ОПТИМАЛЬНА ПОБУДОВА МАТРИЦІ ШАБЛОНІВ ІМОВІРНІСНИХ НЕЙРОМЕРЕЖ У ТЕХНІЧНІЙ ДІАГНОСТИЦІ НА ОСНОВІ ЕКСПЕРТНИХ ОЦІНОК

(стор. 19 – 25)

В. РОМАНУК
Механіко-електричний факультет Військово-морської Академії Польщі, Польща
ORCID: https://orcid.org/0000-0003-3543-3087

У сфері технічної діагностики багато завдань вирішуються за допомогою автоматизованої класифікації. Для цього найкраще підходять такі класифікатори, як імовірнісні нейронні мережі, завдяки своїй простоті. Для отримання ймовірнісної матриці шаблонів нейронної мережі для технічної діагностики зазвичай використовують експертні оцінки або вимірювання. Матрицю закономірностей можна вивести прямо шляхом простого усереднення цих оцінок. Однак середні значення не завжди є найкращим способом обробки експертних оцінок. Мета – запропонувати метод оптимального виведення матриці закономірностей для технічної діагностики на основі експертних оцінок. Основним критерієм оптимальності є максимізація продуктивності, до якої входить підкритерій максимізації швидкодії. Перш за все, визначається максимальна ширина матриці візерунка. Ширина не перевищує кількості експертів. Потім для кожного стану об'єкта експертні оцінки кластеризуються. Кластеризацію можна здійснити за допомогою методу k-середніх або подібного. Центроїди цих кластерів послідовно утворюють матрицю шаблону. Оптимальна кількість кластерів визначає оптимальність імовірнісної нейронної мережі шляхом максимізації її продуктивності. Загалом, більшість результатів відсотка частоти помилок імовірнісних нейронних мереж, здається, майже експоненціально зменшуються зі збільшенням кількості кластеризованих експертних оцінок. Тому, якщо оптимальна кількість кластерів визначає занадто «широку» матрицю шаблонів, швидкість роботи якої нестерпно низька, максимізація продуктивності передбачає компроміс між мінімальним відсотком помилок і максимально допустимою повільністю швидкості роботи ймовірнісної нейронної мережі. Оптимальна кількість кластерів визначається при асимптотично мінімальному відсотку частоти помилок або при прийнятному відсотку частоти помилок, який відповідає максимально допустимій повільності швидкості роботи. Оптимальність практично відноситься до одночасної прийнятності частоти помилок і швидкості роботи.

Ключові слова: технічна діагностика, ймовірнісна нейронна мережа, матриця патернів, експертні оцінки, кластеризація, максимізація продуктивності.

УДК 004.855.5
DOI: https://doi.org/10.20535/2708-4930.2.2021.244188

ВЕКТОРНО-ПРОСТІРНІ МОДЕЛІ ПЕТИЦІЙ МІСТА КИЄВА

(стор. 26 – 34)

Р. ШАПТАЛА
Навчально-науковий комплекс «Інститут прикладного системного аналізу»
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: https://orcid.org/0000-0002-4367-5775

Г. КИСЕЛЬОВ
Навчально-науковий комплекс «Інститут прикладного системного аналізу»
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: https://orcid.org/0000-0003-2682-3593

У цьому дослідженні ми досліджуємо та порівнюємо два способи створення векторної просторової моделі петицій міста Києва. Обидві моделі побудовані на основі векторів слів на основі гіпотези розподілу, а саме Word2Vec і FastText. Ми навчаємо вектори слів на наборі даних петицій міста Києва, попередньо обробляємо документи та застосовуємо усереднення для створення векторів петицій. Візуалізації векторних просторів після зменшення розмірності через UMAP демонструються в спробі показати їх загальну структуру. Ми показуємо, що отримані моделі можна використовувати для ефективного запиту семантично пов’язаних петицій, а також для пошуку кластерів пов’язаних петицій. Проаналізовано переваги та недоліки обох моделей.

Ключові слова: модель векторного простору, FastText, Word2Vec, аналіз петицій, UMAP.

УДК 004.383
DOI: https://doi.org/10.20535/2708-4930.2.2021.244191

ВИДІЛЕННЯ МІСЦЕВИХ ХАРАКТЕРИСТИК НА ЗОБРАЖЕННЯХ

(стор. 35 – 47)

П. СЕРГІЄНКО
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: https://orcid.org/0000-0003-3030-0074

А. СЕРГІЄНКО
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: https://orcid.org/0000-0001-5965-1789

М. ОРЛОВА
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: https://orcid.org/0000-0002-6617-4631

Проаналізовано методи виділення локальної ознаки. Аналіз показує, що найбільш ефективні детектори засновані на визначенні градієнта яскравості. Зазвичай використовують детектор кутів Харріса, який є складним у розрахунках. Мінімізація складності алгоритму суперечить як ефективності детектора, так і високому динамічному діапазону аналізованого зображення. Як наслідок, високошвидкісні методи не могли розпізнати характерні точки в умовах сильної яскравості.

Запропоновано модифікацію алгоритму стиснення зображень із широким динамічним діапазоном (HDR) на основі методу Retinex. Він містить адаптивний фільтр, який зберігає краї зображення. Фільтр базується на наборі детекторів ознак, які виконують перетворення

Харріса-Лапласа, яке є набагато простішим, ніж детектор кутів Харріса. Розроблено прототип відеокамери HDR, яка забезпечує чітке зображення. Його структура спрощує конструкцію двигуна штучного інтелекту, який реалізується в FPGA середнього або великого розміру.

Ключові слова: FPGA, виділення ознак, HDR, розпізнавання образів, штучний інтелект.

УДК 004.383
DOI: https://doi.org/10.20535/2708-4930.2.2021.244189

АПАРАТНІ КОМПРЕСОРИ ЗОБРАЖЕНЬ GIF

(стор. 48 – 55)

І. МОЗГОВИЙ
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: https://orcid.org/0000-0001-5469-486X

А. СЕРГІЄНКО
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: https://orcid.org/0000-0001-5965-1789

Р. ЄРШОВ
Кафедра електроніки, автоматики, робототехніки та мехатроніки
Національний університет «Чернігівська політехніка», Україна
ORCID: https://orcid.org/0000-0002-0267-2906

Підвищення вимог до передачі та зберігання даних зараз є одним із актуальних питань. Існує кілька способів високошвидкісної передачі даних, але вони задовольняють обмежені вимоги до їх вузьконаправленої конкретної мети. Підхід до стиснення даних дає вирішення проблем високошвидкісної передачі та зберігання невеликих обсягів даних. Ця стаття присвячена стисненню GIF-зображень за допомогою модифікованого алгоритму LZW з деревоподібним словником. Це призвело до зменшення часу пошуку та збільшення швидкості стиснення даних, і, в свою чергу, дозволяє розробити метод побудови апаратного прискорювача стиснення під час майбутніх досліджень.

Ключові слова: FPGA, GIF, стиснення без втрат, стиснення зображень, словник, апаратне прискорення.

УДК 004.414.22
DOI: https://doi.org/10.20535/2708-4930.2.2021.247770

АРХІТЕКТУРНЕ ПРОВЕДЕННЯ ТА КОНЦЕПТУАЛЬНА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ФАКУЛЬТЕТУ «КПІ-КОННЕКТ»

(стор. 56 – 63)

І. КЛИМЕНКО
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: https://orcid.org/0000-0001-5345-8806

Ю. БУЦЬКИЙ
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: https://orcid.org/0000-0003-1358-2583

К. ГРИЩЕНКО
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: https://orcid.org/0000-0002-8186-960X

М. СІВАЧЕНКО
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: https://orcid.org/0000-0001-8661-1563
В. КРИВЕЦЬ
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: https://orcid.org/0000-0002-6686-0092

Д. КРИВОШЕЙ
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: https://orcid.org/0000-0003-2236-3710

Д. НГУЕН
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: https://orcid.org/0000-0001-8254-3540

Дана стаття присвячена розробці моделі інформаційної системи автоматизації навчального процесу на базі факультету інформатики та обчислювальної техніки НТУУ «Київський політехнічний інститут імені Ігоря Сікорського». Вивчено існуючі освітні системи різних вищих навчальних закладів; визначено основні реалізовані функції подібних платформ. У результаті дослідження отримано модель, яка дозволяє вводити дані про студентів, викладачів та інших співробітників університету, зберігати персональні дані та інформацію про наукові праці користувачів, а також здатна інтегруватися в існуючий інформаційний простір університету.

Ключові слова: навчальний процес, інформаційна система, автоматизація, практичне використання.

УДК 004.383
DOI: https://doi.org/10.20535/2708-4930.2.2021.251693

МІКРОКОНТРОЛЕР ДЛЯ ЛОГІЧНИХ ЗАДАЧ
(стор. 64 – 72)

М. СЕРГІЄНКО
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: http://orcid.org/0000-0001-5965-1789

О. МОЛЧАНОВ
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: https://orcid.org/0000-0001-8384-0918

М. ОРЛОВА
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна
ORCID: https://orcid.org/0000-0002-6617-4631

Запропоновано нову архітектуру мікроконтролера SM16, яка призначена для логічно інтенсивних програм у програмованій вентильній матриці (FPGA). Мікроконтролер має стекову архітектуру, яка забезпечує реалізацію більшості команд за один такт. Короткі, але швидкі програми виводяться завдяки 16-бітним інструкціям, які кодують до трьох незалежних операцій, і інтенсивному використанню стилю потокового коду. Розроблено структуру, яка компілює програму, моделює її та перекладає на ПЗП. Розроблене ядро SM16 з додатковими блоками з трьох стеків, хеш-таблицею та інструкціями, які прискорюють виконання операцій синтаксичного аналізу, використовується для ефективної обробки XML-документів і може часто переналаштовуватися відповідно до набора граматики документа. Швидкість розбору дорівнює одному байту за 24 такти.

Ключові слова: VHDL, XML, парсер, FPGA, стековий процесор, граматика, FSM