Software Development Technology with Automatic Configuration to Classes of Image Processing Problems

A. Nedzved[a,c], I. Gurevich[b], Yu.Trusova[b], and S. Ablameyko[c]

[a] *Joint Institute of Informatics Problems, Belarusian National Academy of Sciences, ul. Surganova 6, Minsk, 220012 Belarus*

[b] *Dorodnicyn Computing Center, Russian Academy of Sciences, ul. Vavilova 40, Moscow, 119333 Russia*

[c] *Belarusian State University, ul. Bobruiskaya 5, Minsk, 220006 Russia*

e-mail: NedzvedA@tut.by, igourevi@ccas.ru, ytrusova@ccas.ru, ablameyko@bsu.by

Abstract—A scheme is proposed to develop image processing and analysis software based on the generation of thesaurus tables in the interpreter's kernel. The scheme is based on the combined features of dynamic libraries and an interpreter with a set of functions for image processing. As a result, the program can be divided into two parts: the first is aimed at professional software developers, and the second at users. The interpreter is able to use extra functions of dynamic libraries. It makes it possible to change the properties of the software without compilation. On the other hand, users can change the graphical interface to improve the convenience of the workplace.

Keywords: image processing, image analysis, thesauri, ontologies, automated analysis of tissue specimen.

DOI: 10.1134/S1054661813020120

INTRODUCTION

Today, such a field of human knowledge as computer engineering involves many different directions that affect software features. As a rule, software support has specific requirements for the organization of programs, depending on active tasks, user skills, and workplace design. Thus, modern requirements on image analysis software require the dynamic organization of functions, user interface, data management for methods of image processing, etc.

Any software can be divided into compiled programs and interpreters. Sometimes a program is represented as a combination of these options. Such software should have a kernel that includes the following components: GUI (graphical user interface), basic functions and commands, and different types of data. The interpreter is commonly used to preserve the history of operations applied to the processing of new images of the same type. In this case, it makes it possible to create additional simple functions based on kernel features without changing software [1]. We propose taking advantage of the interpreter as a nucleus for creating software. In this case, the interpreter acts as a manager of actions and events. It supports the implementation of functions, calculations, and GUI events. In this case, it is possible to change the software design without compilation. On the other hand, complex functions and calculations are performed in external statically compiled modules. This saves computing speed at the same level as in compiled software. Today, such software architecture organization is used for web development and in games. It is called "open software architecture" [2].

Any intelligent application studies objects in terms of their origin, structure, functions, and the area of the problem being solved (e.g., medical problems [3]). Practical solutions with a specific area of application are critical in shaping the program and include

(1) Identification of the type of objects and their main classes.

(2) Analysis of object interactions and reactions required in the formulation of a problem.

(3) Detection of scenes to create a general picture of the problem.

(4) The practical context in a scene towards the solution of the problem.

A necessary condition for achieving these objectives is the systematization, structuring, and, in a sense, formalization of knowledge in data processing, analysis, and pattern recognition. It is known that one of the effective ways to represent knowledge is domain ontology. Ontologies are used to express a formal machine-interpretable description of the semantics of data of a field of knowledge and are a suitable tool for knowledge representation used to extract information from images.

Image analysis is an extremely rapidly growing sector of computer science, and therefore its conceptual structure is changing dynamically. On the other hand, the efficiency of research in image analysis and solving applied image analysis problems depends to a large extent on the standardization and formalization of the

Received October 16, 2012

used descriptions and representations of both images and descriptions of methods of their processing, analysis, and recognition. Thesaurus representation of image analysis domain can be used as standardization and formalization tool, as well as to provide skilled and unskilled users solving image analysis problems, with access to image analysis knowledge, e.g., by creating an effective universal image analysis program. It is important here that the thesaurus for image analysis and the ontology constructed on its basis will make it possible to use convenient and effective procedures to access standardized and structured representations of that knowledge.

In the most general definition, a thesaurus is a terminological resource consisting of a glossary of terms used to describe the concepts of a domain with specified relationships between terms.

Actions performed in image analysis are regular and depend on an assessment of image quality and objects in it. The correspondence of image features and image processing functions is performed by the thesaurus as a list of matching processing operations and image analysis rules. In this paper, medical images were selected for analysis. Modern computer systems supporting the analysis of medical images have different sets of functions and interfaces. At the same time, the most commonly used methods for analyzing medical images operate on standard algorithms that make it possible to obtain good results. The group of morphological characters used to detect medical objects in such systems usually has no links to the functions [3]. This focuses systems to address specific problems.

This paper proposes a scheme for the development of software for images processing and analysis based on the generation of thesaurus tables in the interpreter's kernel. The scheme is based on combined features of the dynamic libraries and interpreter with a set of functions for image processing. As a result, the program can be divided into two parts: the first part is aimed at professional software developers, and the second part, at users. The interpreter is able to use the extra functions of dynamic libraries. This makes it possible to change the properties of the software without compilation. On the other hand, users can change the graphical interface to improve the convenience of the workplace.

DEVELOPMENT OF THE STRUCTURE OF THE SET OF DESCRIPTORS FOR IMAGE ANALYSIS

Any domain is based on a system of concepts. The simulation of a domain is a difficult task because of the need to combine individual conceptual models of the world by different experts, organizations, communities, etc., and to create a single model that would not cause significant discord. Elements of the domain model are concepts, links (relationships) between concepts, and properties of these concepts and relationships. Relationships are used to identify a particular concept in the context of other concepts, and properties are used to further clarify the characteristics of the concept. In general, the construction of the domain model makes it possible to develop a common understanding of the new field of knowledge and to identify common vocabulary (terminology) to be shared within a particular community.

Recently, researchers in the field of computer science and, in particular, in the area of artificial intelligence, have been actively using the term "ontology," which has been borrowed from philosophy.

One of the first definitions of "ontology" in the context of conceptualization and representation of knowledge was given in [4]: "Ontology defines the basic terms and links (relationships) between the terms to form the vocabulary of the domain and rules of combination (association) of terms and links (relationships) to expand the vocabulary." In this paper, a thesaurus representation of the ontology of the domain of image analysis is used.

One of the main objectives of creating a thesaurus is to develop its structure. For each concept of the domain, a set of terms used to describe this concept is assigned in the thesaurus. One of these terms (the preferred one) is chosen as a descriptor, and the rest are ascriptors (synonyms).

The structure of the set of descriptors for image analysis is determined by the state of the art of image analysis and recognition theory, experience in solving applied problems, functional requirements, and the specifics of the lexical content of the domain language. It includes the following basic elements [5]:

(1) Topics (thematic sections) of descriptors.

(2) Functional categories of descriptors.

(3) Set of relationships between descriptors.

(4) Scheme of an entry.

The development of any thesaurus includes the definition of the thematic coverage. The following areas are defined, in accordance with which thematic sections of vocabulary are formed in a set of descriptors for image analysis:

(1) Terminology describing the image itself.

(2) Image processing.

(3) Image analysis.

(4) Image recognition.

(5) Pattern recognition.

(6) Applied problems.

(7) General mathematical terminology.

Based on the specifics of the subject area of "Image Processing, Analysis, Recognition, and Understanding," the following categories of terms are included in a set of descriptors for image analysis.

(1) Objects Category:

1. Names of images (e.g., binary image, raster image, 2D image, quantized image, etc.).

2. Names of elements of images (e.g., edge, region, pixel, etc.).

(2) Task Category:

1. Names of image processing tasks (e.g., image enhancement, image restoration, image quantization, etc.).

2. Names of image analysis tasks (e.g., image segmentation, texture analysis, etc.).

3. Names of pattern recognition tasks, including the names of the tasks of image recognition (e.g., cluster analysis, error estimation, etc.).

3) Approaches Category:

1. Names of approaches to image processing (e.g., wavelet-based image processing, etc.).

2. Names of approaches to image analysis (e.g., model-based image analysis, etc.).

3. Names of approaches to pattern recognition (e.g., statistical pattern recognition, etc.).

(4) Methods Category. It includes

1. Names of image processing methods (algorithms) (e.g., fractal image compression, equal interval quantization, etc.).

2. Names of image analysis methods (algorithms) (e.g., structural texture description method, adaptive edge detection, etc.).

3. Names of pattern recognition methods (algorithms), including the names of image recognition methods (e.g., image algebra-based technique, EAC-based technique, etc.).

(5) Tools Category:

1. Names of image processing tools (operators, transformations, and filters) (e.g., median filter, discrete Fourier transform, etc.).

2. Names of image analysis tools (e.g., Prewitt edge detector, Sobel edge detector, etc.);

3. Names of tools of pattern recognition, including the names of image recognition tools (e.g., maximum likelihood decision rule, cluster assignment function, etc.).

(6) Characteristics Category:

1. Names of tool characteristics (e.g., threshold, structuring element, convolution kernel, etc.).

2. Names of image characteristics (element descriptions) (e.g., brightness, color model, contrast difference, etc.).

One of the main differences of a thesaurus from other dictionaries, such as glossaries, is that basic semantic relationships between terms are indicated in the thesaurus. In addition to standard types of relationships, i.e., equivalence, hierarchy, and associativity, for image processing the necessity was established to include into the thesaurus the following links between specific to the subject area descriptors:

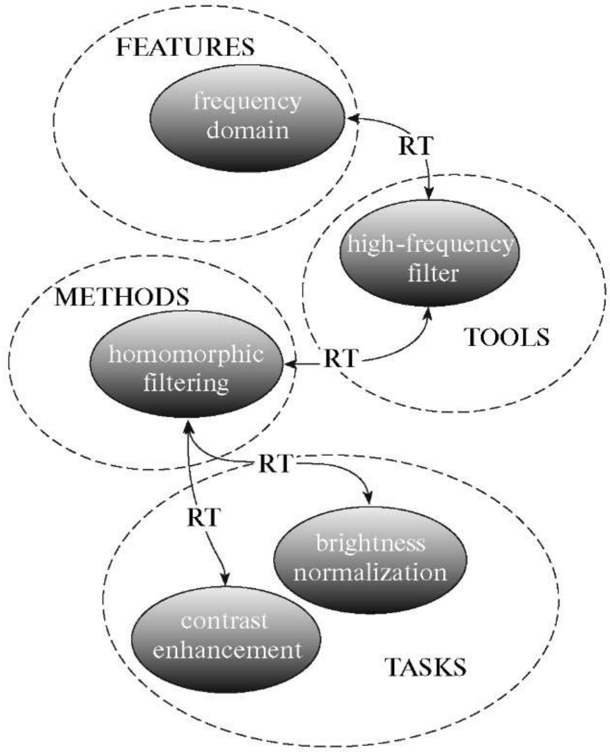

Fig. 1. Classification scheme of "homomorphic filtering" concept.

(1) Task–Method, e.g., image segmentation–edge-based image segmentation.

(2) Method–Tool, e.g., gradient-based edge detection–Sobel edge detector.

(3) Object (image view)–Method, e.g., binary image–binary noise reduction.

(4) Tool–Result, e.g., edge detection operator–edge image.

(5) Tools–Tool characteristic, e.g., image opening operation–structuring element.

Fig. 1 shows an example of a classification scheme of the concept *homomorphic filtering* illustrating associations between different categories of descriptors.

A more detailed description of the developed thesaurus and ontology based on it can be found in [5, 6].

GENERAL SCHEME FOR PROCESSING HISTOLOGIC SPECIMEN IMAGES

The processing of medical images can be divided into several stages

1. Input of image and improvement of its quality.

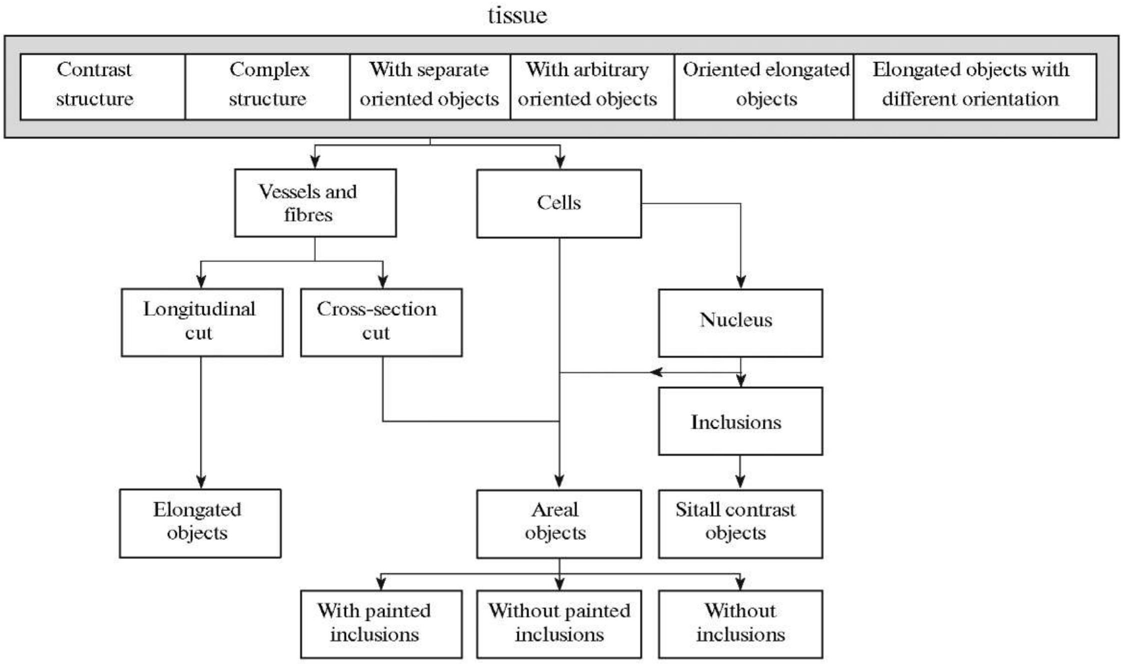

Fig. 2. Hierarchical structure of histological tissue.

2. Segmentation.

3. Identification of objects.

4. Measurement of objects and calculation of their characteristics.

5. Analysis.

Each step consists of different sequences of functions. Their use depends on the properties of the image or its estimates, which can be determined in most cases, e.g., evaluation of contrast, noise, or blur. We can construct a table of image processing functions and evaluations of image properties.

For example, the use of segmentation methods for a histological image depends on many conditions. Typically, an image is divided into separate areas for analysis of histological specimens. Thus, the segmentation process (e.g., determination of homogeneous areas in the image) is considered a major step for a formal description of the scene. This is necessary to determine the right set of functions and their characteristics in order to choose the best segmentation method.

The required histological objects are defined according to the tasks to be solved [7]. Automatic analysis of histological specimens can be done based on the topological features of the image. This makes it possible to define the entire procedure for the study of its origin. However, automatic analysis in histology depends on optical zoom. In each magnification, there is a definite group of topological features of tissue and its components, so it is better to conduct such an analysis for different levels of magnification.

Figure 2 shows the general scheme of the hierarchical analysis of objects for histological images. Different fragments of tissues composed of groups of homogeneous cells and fibers form the whole image of the histological specimen. As a rule, these objects represent a certain texture. Thus, algorithms of growth of areas are popular methods for its allocation.

The assessment of features of the image makes it possible to determine the functions necessary for its processing.

As a result, a table of relations of functions and evaluations of image characteristics is constructed (Fig. 3).

At each implementation step, priorities are defined for functions. For example, in order to improve the image, impulse noise is removed and the next priority level will be assigned to functions to improve contrast and correct object borders. Priorities determine the order of execution of functions and the need for additional analysis of image features.

Image processing functions are among the main topics of computer vision. It corresponds to their use for changing the image. Each function changes the properties of the image and is used for specific processing. Properties and function features should be described in the global table of the interpreter and be accompanied by additional information.

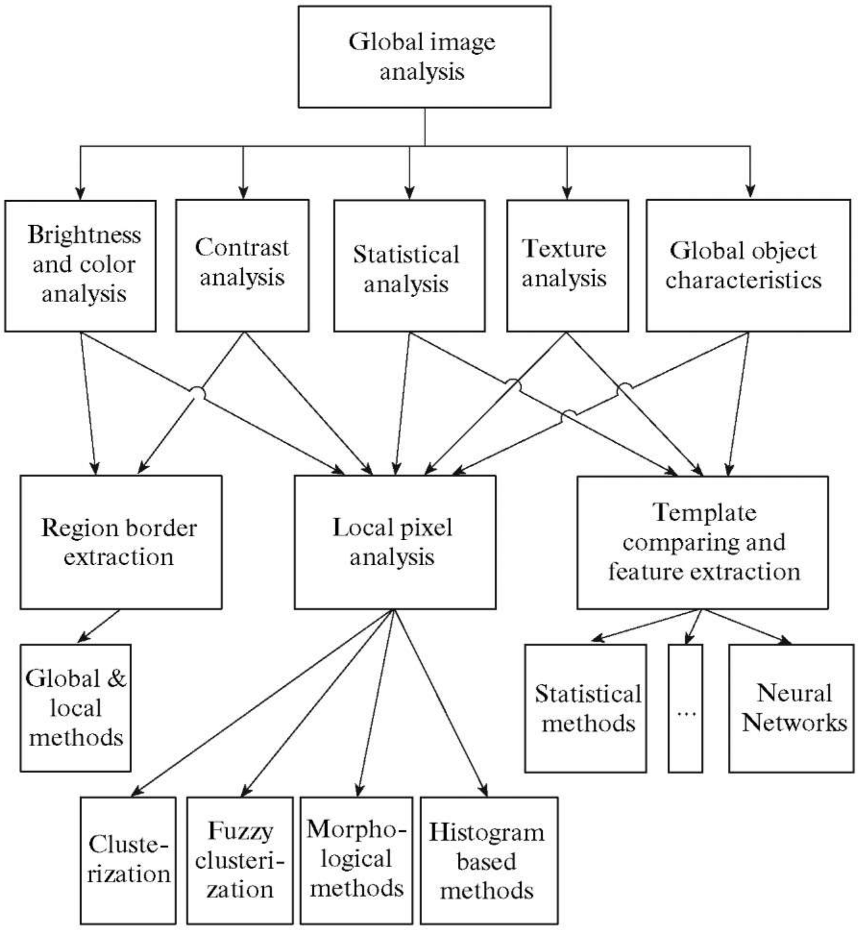

Fig. 3. General scheme of hierarchical analysis of histological images.

4. DEVELOPMENT OF SOFTWARE FOR HISTOLOGICAL IMAGE ANALYSIS

In order to develop the block diagram of the software interface and architecture, the functionality and compatibility of available software development tools are assessed.

Based on materials on the website of Guillaume Marceau [1], who in his study used 72 parameters to assess implementations of programming languages and compared them on the basis of 19 special tests, the Lua interpreter was chosen as the kernel [8].

Our system is based on the Lua language interpreter. It is used as a basic module and provides the interaction between complex components. A GUI and global variables and structures of the image were additionally included into the kernel of the interpreter. The GUI architecture is implemented by combining the Highgui library from the OpenCV package [9] and Qt libraries [10]. Image processing and analysis functions are supported by means of communication of the OpenCV library with Lua, which is implemented in a separate library.

The structure of the image is defined in a GUI module based on the OpenCV library, which is responsible for the visualization and presentation of images. Headlines of the image structure are defined as global pointers of the Lua interpreter and have a special type, i.e., for user data. This type (Userdata) corresponds to indicators of the address range of the computer. This module also includes functions for image reading/writing and simple functions for image processing and selection of the interactive contour. All interactive functions return values to a separate table of interpreter data, changing its state. For tasks of monitoring of various diseases, it is possible to use multiple modules simultaneously. In this case, the interaction is carried out using global variables of the Lua interpreter and properties of user data of interpreter type.

During image processing with the Lua kernel, basic sets of instruction sequences are used; however, during the processing instructions in the script can be changed, which affects the result of image analysis. Thus, the entire process of processing and image anal-

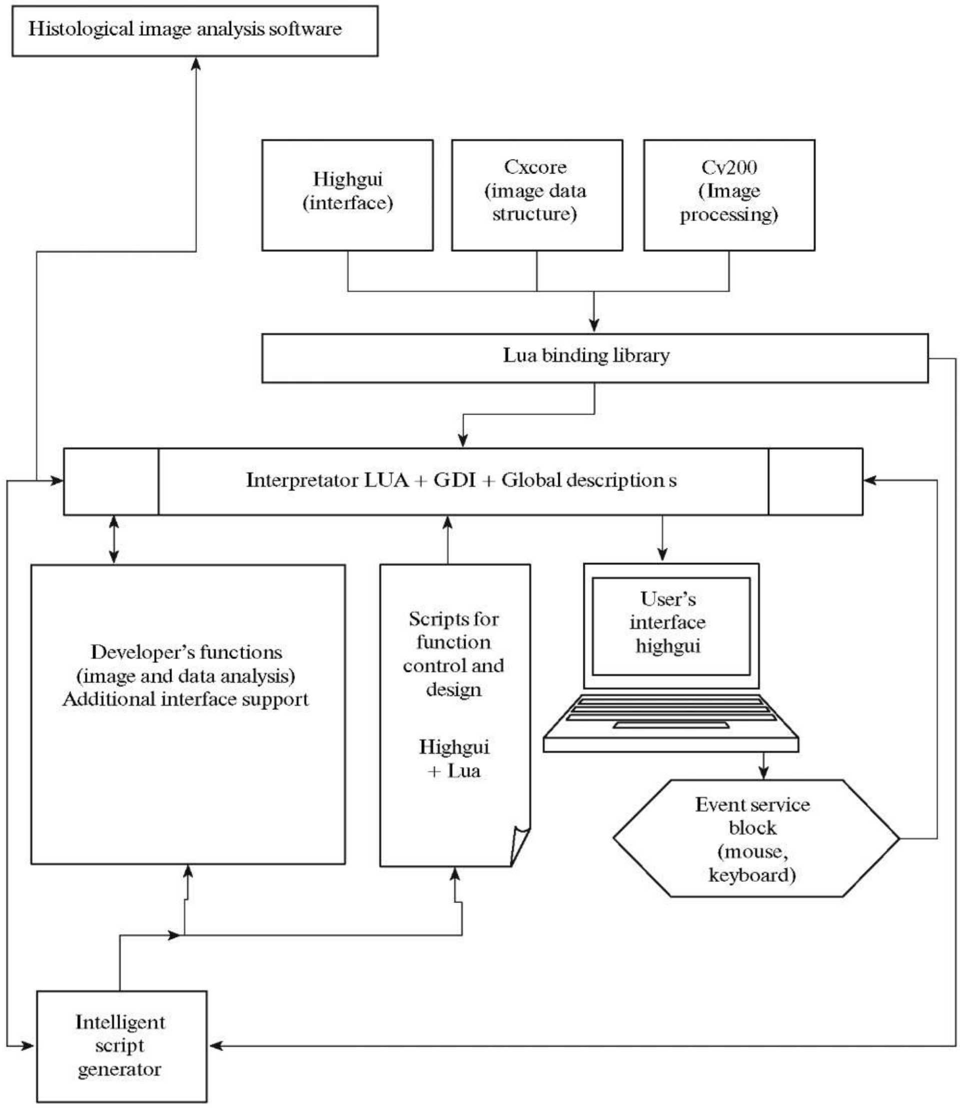

Fig. 4. Scheme of intelligent software with self-programming capabilities.

ysis is formed by several sets of scripts connected by nodes, providing results dependent on particular conditions of the problem. The communication between nodes in the same scenario is implemented by direct function calls by event-slots or by a virtual explosion of the method for its first entry. Adding new advanced functions to processing and analysis is carried out by a group of developer functions. New functions are determined by variables of the global table of the Lua interpreter.

The internal control of software is conducted by Lua text scripts, which are divided into two categories

(1) Scenarios for the analysis of sequences of images.

(2) Scenarios for operational functionality and configuration of software on user workstations.

All scenarios–scripts are stored in text format and are easily accessible; however, they cannot be modified only by developers and users do not have such rights. Software management scenarios make it possible to create new advanced functions for image analysis and processing.

The script generation module defines new features of the program as a set of simple functions from the libraries described above. It includes an intelligent component for associating the image processing results with the characteristics. Of course, this option can be implemented only for specific tasks. In our case, problems were focused on histological analysis of images. Each interpreter defines the functions according to a particular table. In our scheme, such a table is used later to determine the relationship between characteristics of the image with the functions of the program included in the control script. As a result, the software acquires intellectual self-programming possibilities.

The script generation module consists of two parts, i.e., image analysis evaluation and generation itself. The determination of the structure of the script begins with a global analysis of images, which includes specification of the following characteristics: the description of the histogram and main characteristics of the statistical analysis of pixel distribution, fractal evaluations, and description of the texture. Based on these estimates, necessary sequences of functions for image processing are formed. For example, evaluations obtained in analyzing image noise and blur are used to determine the necessary filter sets that affect the overall quality of the image. After comparing estimates from a broader global table of the interpreter, a script is generated for image preprocessing in order to enhance its quality.

Local image analysis is performed using convolution with different filters and statistical analysis of linear profiles of image fragments. Such an analysis makes it possible to estimate characteristic cell borders and contrast. This assessment determines the functions to enhance image contrast and underscore boundaries. After the image is changed, the result is again evaluated. If the image is of poor quality after image analysis and determination of the correction function, the procedure should be repeated. As a result, a robust script is formed that can be changed by the user or developer if necessary. This scenario is only a suggestion to improve image quality and determine the user control parameters of the interface. The same mechanism is used for image analysis by the function that determines the phases of image segmentation and postprocessing.

As a result, the program generates sets of scripts that correspond to the function for selecting desired objects. It can be used to determine histological objects in an image. Scenarios for selecting kernels and membranes in histological images were developed for our program. Then the characteristics of objects are calculated. This is necessary to identify types of objects that are represented in the image. Initially, objects are divided into five main types of structures: areal, front, needles, dendrites, and networks. This object recognition procedure begins to execute a cascade of operations on the selection of objects. The direction and organization of this cascade depends on the type of tasks. The use of global fractal and textural characteristics is the first step in detecting the geometrical type and shape of the description of the object. Then different characteristics and their combinations are used in the cascade.

In order to determine the image processing function in the scenario, we tried to use a Kohonen neural network [11, 12]. For this class of neural networks, the key element is the Kohonen layer. It consists of adaptive linear combiners. Assessments of global conditions of the image are used as weights in the neural network. The adjustment of input weights and the network signal quantization vector are closely related to the cluster analysis algorithm (e.g., by dynamic kernels or K-means).

The script generation module uses Lua metatables of functions for image processing with added image evaluation information. Such tables of function determination are implemented in the Kohonen neural network as sets of weights. As a result, the module generates scenarios for various image processing tasks in the form of text files. Users can analyze and change these scenarios.

7. CONCLUSIONS

The technology of software development based on automatic generation of scripts, which include a set of simple image processing functions for different problems of analyzing histological tissue. This software is based on open architecture principles, which make it possible to change the design in real time without compilation at the user's workplace. On the other hand, the program run time is the same as in the compiled version. The proposed software architecture facilitates the development of programs for medical image analysis, in particular, histological analysis.

The use of such principles in the development of programs can significantly facilitate the development of software. This is because packages developed under the open architecture principle pass two development stages: the compiled and interpreter stage. As a result, the cycle of life and growth of the software is extended considerably. In addition, the dynamic parameters of the program are significantly improved, because when interpreter components are used, the program can be easily modified and adapted to new challenges. However, in order to implement such a modification, special knowledge is required. The use of an intelligent agent equipped with appropriate means of relationships between descriptors and functions in this scheme allows for self-modification of the software or by a regular user without the help of a specialist. The technology was successfully tested for histological image analysis problems. In the future, it will be adapted to other problems related to image processing. This technology will be developed according to the scenario from the construction of the concept of the problem to the formation of the user interface and the algorithm for solving the problem.

The results of applying this technology can impact not only the concept of the software, but also the progress of clinical diagnostics. The following results of this work are important in the diagnostics and monitoring

(1) Improvement in diagnostics quality and reduction in the risk of false diagnosis due to automation of age peculiarities.

(2) Faster early diagnosis of neoplastic processes in organs.

(3) Improvement in quality of monitoring primary and residual tumors in organs.

(4) Optimization of stratification of treatment based on analysis of characteristics of volume-structural measurements of a tumor in the treatment process.

(5) Introduction of objective methods for automating diagnosis and monitoring neoplastic processes in organs by processing of 2D and 3D images into other medical facilities and diagnostic systems.

Currently, this technology has been implemented in the QTIP pilot program complex, which is used for solving problems of disease diagnosis using histological images. In the future, it will be adapted for nanoscale, satellite, and aerial survey images. In order to achieve this goal, it is necessary to create new descriptive attributes, which should be associated with image processing functions.

In addition, the overall design of this software will certainly manifest itself in the short term in constructing complex software systems.

ACKNOWLEDGMENTS

This work was supported in part by the Russian Foundation for Basic Research (project nos. 11-01-00990 and 12-07-31123) and the Basic Research Program of the RAS Presidium “Information, Control, and Intelligent Technologies and Systems” (project no. 204).

REFERENCES

1. G. Marceau, “The Speed, Size and Dependability of Programming Languages. Blog “Square Root of x Divided by Zero”. http://gmarceau.qc.ca/blog/2009/05/speed-sizeand-dependability-of.html
2. G. Reitmayr and D. Schmalstieg, “An Open Software Architecture for Virtual Reality Interaction,” in “Matrix: a Tool to Visualize Clusters in High Dimentional Data,” Tech. Rep. 36 (Univ. of Marburg, Department of Computer Science, 2003), pp. 1–12.
3. A. Nedzved, A. Belotserkovsky, T. M. Lehmann, and S. Ablameyko, “Morphometrical Feature Extraction on Color Histological Images for Oncological Diagnostics,” in *Proc 5th Int. Conf. on Biomedical Engineering* (Innsbruk, Feb. 14-16, 2007), pp. 3479–384.
4. R. Neches, R. E. Fikes, T. Finin, T. R. Gruber, T. Senator, and W. R. Swartout, “Enabling Ontology for Knowledge Sharing,” AI Mag. **12** (3), 36–56 (1991).
5. V. N. Beloozerov, I. B. Gurevich, N. G. Gurevich, D. M. Murashov, and Yu. O. Trusova, “Thesaurus for Image Analysis: Basic Version,” Pattern Recogn. Image Anal.: Adv. Math. Theory Appl. **13** (4), 556–569 (2003).
6. I. B. Gurevich, O. Salvetti, and Yu. O. Trusova, “Fundamental Concepts and Elements of Image Analysis Ontology,” Pattern Recogn. Image Anal.: Adv. Math. Theory Appl. **19** (4), 603–611 (2009).
7. T. Kanade, Z. Z. Yin, R. Bise, S. Huh, S. Eom, M. F. Sandbothe, and M. Chen, “Cell Image Analysis: Algorithms, System and Applications,” in *Proc. of WACV11* (Kona, 2011), pp. 374–381.
8. R. Ierusalimschy, *Programming in Lua the 2nd* (Lua.org, March 2006).
9. G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library* (O’Reilly, 2008).
10. J. Blanchette, *Summerfield M. C++ GUI Programming with Qt 4*, 2nd ed. (Prentice Hall, 2008).
11. S. Kaski, “Data Exploration Using Self-Organizing Maps,” in *Acta Polytechnica Scandinavica, Mathematics, Computing and Management in Engineering Series* (Espoo, 1997), No. 82.
12. Ultsch, 2003 U*-Matrix: a tool to visualize clusters in high dimentional data. University of Marburg, Department of Computer Science, Technical Report Nr. 36: 2003. P. 1-12.

Translated by O. Pismenov

Alexander Nedzved. Graduated from the Belarus State University in 1992. He is deputy head of the laboratory of Image processing and recognition of United Institute of Informatics Problems of Belarussian Academy of Sciences. He also works in Belorussian State University and Belorussian state medical university. With workgroup he has created a image analysis system “BIOSCAN” (Minsk) (http://www.itlab.anitex.by/bioscan/), a image analysis system “IMAGEWARP”, cytology image processing system “Cytron”, system of magnetooptical films analysis “Zubr”. Scientific interests are image processing, feature extraction, algorithms of 2D–3D image thinning and segmentation and 3D reconstruction, segmentation of colour images, pattern recognition, mathematical morphology, knowledge-based systems, intelligent software. Member of Belarus Association for Image Analysis and Recognition. Author of more then 100 publications. (http://nedzveda.narod2.ru)

Igor B. Gurevich. Born 1938. Dr. Eng. (Diploma Engineer (Automatic Control and Electrical Engineering), 1961, Moscow Power Engineering Institute, Moscow, USSR); Dr. (Theoretical Computer Science/Mathematical Cybernetics), 1975, Moscow Institute of Physics and Technology, Moscow, USSR. Head of department at the Dorodnicyn Computing Centre of the RAS, Moscow; assistant professor at the Faculty of Computational Mathematics and Cybernetics, Moscow State University. Since 1960, has worked as an engineer and researcher in industry, medicine, and universities and in the Russian Academy of Sciences. Area of expertise: image analysis; image understanding; mathematical theory of pattern recognition; theoretical computer science; pattern recognition and image analysis techniques for applications in medicine, nondestructive testing, and process control; knowledge bases; knowledge-based systems. Two monographs (in coauthorship); 135 papers on pattern recognition, image analysis, and theoretical computer science and applications in peer-reviewed international and Russian journals and conference and workshop proceedings; one patent of the USSR and four patents of the RF. Scientific secretary of the National Committee for Pattern Recognition and Image Analysis, member of the governing board of the International Association for Pattern Recognition (representative from the Russian Federation), IAPR fellow. Has served as PI of many research and development projects as part of national research (applied and basic) programs of the Russian Academy of Sciences, the Ministry of Education and Science of the Russian Federation, the Russian Foundation for Basic Research, the Soros Foundation, and INTAS. Vice Editor–in–Chief of "Pattern Recognition and Image Analysis", International Academic Publishing Company "NAUKA/INTERPERIODICA" Pleiades Publishing.

Yulia O. Trusova. Born 1980. Graduated from the Faculty of Computational Mathematics and Cybernetics of Lomonosov Moscow State University in 2002. Received a Ph.D (Theoretical Foundations of Computer Science) in 2009. Works at the Dorodnicyn Computing Centre of the RAS (Moscow) as a senior researcher of the Department of Mathematical and Applied Methods of Image Analysis and Nonlinear Tasks. Scientific interests are mathematical theory of pattern recognition and image analysis, knowledge bases, knowledge-based systems, ontology development. Coauthor of more than 40 papers. Member of the National Committee of the RAS for Pattern Recognition and Image Analysis.

Sergei V. Ablameiko. Graduated from Belarusian State University (BSU) with degree in mathematics. From 1978 to October 2008 worked at United Institute of Informatics Problems of National Academy of Sciences of Belarus (up to August 2002 Institute of Technical Cybernetics). Defended candidate's thesis1984; doctoral thesis 1990. Awarded title of professor 1992. Elected corresponding member of National Academy of Sciences of Belarus 2004. Appointed member of Presidium of National Academy of Sciences by presidential decree 2004. January 1, 2005, appointed Academician Secretary of Physics, Mathematics, and Informatics of NAS of Belarus (secondary position). For a number of years, a professor of Department of Mathematical Support of Automated Control Systems of Faculty of Applied Mathematics and Computer Science of BSU (secondary job). October 31, 2008, appointed Rector of Belarusian State University by Presidential Decree ¹ 593. June 5, 2009, elected Academician at General Meeting of the National Academy of Sciences of Belarus. Published more than 350 scientific works, including 11 books (3 of them published in the United States, England, and Poland). Thirty-four collected papers and five brochures have been published under his editorship (seven of them in the Netherlands, Italy, and Poland). More than 180 of his works have been published in English in international journals and proceedings of international conferences. Supervised nine candidate and two doctoral dissertations. In last 19 years, has also worked as a professor at BSUIR (1998–2003) and BSU (1989–1998, 2003–present). Chairman and cochairman of 20 international conferences held in Germany, Italy, Poland, Russia, and Belarus; member of program committees of more than 80 international conferences. Invited repeatedly as speaker. Editor-in-chief of the journal *News of the NAS of Belarus, Series of Physical and Mathematical Sciences* and *Computer Science*. Member of editorial board of international journals *Pattern Recognition*, *Pattern Recognition Letters*, *Machine Graphics and Vision*, *Pattern Recognition and Image Analysis*, and *Journal of Computer and Information Technology*, published in Europe, the United States, and Russia; and the Belarusian journals *Reports of NAS of Belarus* and *Science and Innovation*. In 1998, for his contribution to the theory of image processing, became the first scientist of the CIS and Eastern Europe to be elected Fellow of the International Association for Pattern Recognition (IAPR). In 2004, elected vice-president, and in 2006, became the first vice-president of IAPR. Elected Fellow of British Institute of Electrical Engineers (IEE) (1995), Senior Member of American Institute of Electrical and Electronics Engineers (IEEE) (1995), Chartered Engineer of European Engineering Council (2001), foreign member (academician) of Royal Spanish Academy of Doctors (2009), foreign member (academician) of European Academy of Economics and Management (2010), and Academician of the European Academy (2011).

Awarded Jubilee Sign of Standing Committee of Union State "Ten Years Since the Day of Unification of the Peoples of Russia and Belarus," the Sign of the Russian Aviation and Space Agency "For International Cooperation in Astronautic Science," the Yuri Gagarin Gold Medal of the Cosmonautics Federation of Russia, the Norbert Wiener Bronze Medal of the International Academy of Information Technology for his contribution to the development of computer science, and other awards.