Draft:
Nedzved, A. Intellectual software for analysis of histological images / A. Nedzved, V. Starovoitov // Proc. of WISP 2011 :IEEE 7th Intern. Symposium on Intelligent Signal Processing. – Floriana, 2011. – P. 130–134.

Intellectual software for analysis of histological images

Nedzved Alexander
United Institute of Informatics Problems of the National Academy of Sciences of Belarus
Minsk, Belarus
Nedzveda@tut.by

Starovoitov Valery
United Institute of Informatics Problems of the National Academy of Sciences of Belarus
Minsk, Belarus
valerys@newman.bas-net.by

Abstract— **In this paper intelligent software for analysis of histological images is described. The core of this software is an interpreter, which combines simple image processing functions. Based on analysis of image properties a script is generated for interpreter. It consists of simple functions and is able to solve the complex problem of histological image analysis.**

Keywords-component; image processing and analysis; intelligent software; automatisation of histological analysis

I. INTRODUCTION

Computer engineering includes many different parts that influence to properties of software. Medical software has specifically requirements to software organizations. It depends of condition of solving tasks, knowledge of user's software, and environment of user's work place. Therefore modern requirements to medical software have dynamical organization of functional, user interface and data managing for medical radiological methods, optical microscopy, endoscopy and ophthalmology and etc.

Any software may be divided into compiled programs and interpreters. Sometimes software is represented as a mixture of these variants. Such software for science has a compiled kernel that includes GUI (graphical user interface), basic functions and calculations, different types of data representation. An interpreter usually is used as saved history of operations applied to processing of a new image of the same type. In this case the interpreter allows to create additional simple functions on base kernel possibilities but can not change the software [1]. We propose to use an interpreter as a kernel for our software. In this case the interpreter is used as a manager of action. It supports by performance of the functions, calculations, and GUI events. In this case we can change the software design and organization without a compilation stage. On other side a complex function and calculations are realized in an external public compiled modules. This will keep the speed of calculation such as the compilation software. Today similar software organization is used for a web and game development. It is named an "open software architecture" [2].

Our software is based on a mixture of a compiled library and an interpreter with many image processing functions. In result the software may be divided into two parts: for the professional software developers and software for designers or users. The interpreter has possibilities for including additional functions from a compiled dynamical library. It allows to change software properties and software applications without a compiling stage. On other side users can change the graphical interface for improvement comfort conditions for development and easy evolution of software. We apply such technique for developing histology image analysis software [3].

Modern computer support and facilities in microscopy bring new perspectives in studying of cell structures. At the same time, the most commonly used method for a tissue analysis is still the well known morphological method, which allows to get reasonable biological conclusions after an image analysis. Group of morphological features, which are used for detection of similar types of cells and for an organ and tissue fragment analysis, is noticeably extended. Usually, there is no any relation between different types of features. Therefore, types of histological tissue fragments are separated from each other by their morphological features. Systematization of histological objects is very important in order to provide a morphological analysis and oncologist diagnosis.

There are various approaches to segment biomedical images. One of the most popular of them is based on mathematical morphology. Many morphology based algorithms for cell segmentation have been proposed through the last years [4]. The initial image segmentation is determined by classifying image local variation information obtained by dilation and erosion operations. The median filter may be used to smooth the segmented image. It removes small areas of misclassified pixels while avoiding significant changes to the cell profiles. The erosion operation is finally used to restore the cell areas. An edge-based segmentation may be divided into two independent stages: edge detection and edge linking. The detected edges are used to determine cell locations and a contour model is further used to select the set of edges involved in the cell locations. In Ref. [3], authors have proposed an edge-based potential aimed at the elimination of local minima due to undesired edges. This approach integrates knowledge about features of the desired boundaries apart from gradient strength and eliminates local minima, which make the

segmentation results less sensitive to initial contours. Color is an important feature in the histological image segmentation. There are several effective algorithms for automatic detection of cells and other histological objects [5]. However, these algorithms work under certain conditions to solve particular problems.

II. COMMON PROCESSING SEQUENCE FOR ANALYSIS OF HISTOLOGICAL SAMPLE IMAGES

A. Processing sequence for analysis of histological sample images

Processing of histological images may be divided into several steps:

1. Input and image enhancement;
2. Segmentation;
3. Object detection (identification);
4. Measuring;
5. Analysis.

Every step consists of execution of a set of functions. Application of a function depends on image properties or image estimations. It is possible to define such estimations in many cases, for example, for contrast, noise or blurring. We can construct a table of image processing functions and image estimations.

For example, for histological image applications segmentation methods depend on many image conditions. Usually, an image is decomposed into separated areas to analyze the histological sample. Therefore, the segmentation process (i.e., extraction of homogeneous regions in image) is considered as a basic step for a formal scene description. It is necessary to define a correct set of features and feature characteristics for a suitable choice of segmentation methods.

Histological objects that are to be extracted may be defined according to tasks to be solved. Automated histological specimen analysis is based on topological features of images. It allows to define the whole procedure of study for object extraction. However, automatic analysis of histological specimen depends on the optical magnification of the image. In each magnification there is a certain group of topological features of tissue and its components. This fact has prompted to consider histological objects over magnification of histological specimens.

Figure 1 presents a general scheme of hierarchical analysis of objects in histological images. Different tissue fragments, which are composed of a group of homogeneous cells and fibers, form an entire image of a histological sample. Usually these objects are represented by a certain texture. Therefore, a region growing approach is used for object extraction.

From initial image conditions it is possible to define a function for image processing and analysis.

In result, a table of connection function and image estimation are constructed (fig.2).

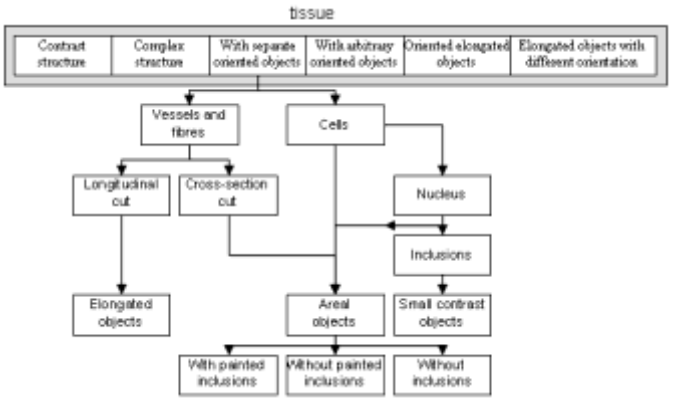

Fig. 1. General scheme of hierarchical analysis of histological images.

In each step, functions are indicated by priorities. For example, for the image improving step the higher priority is defined for noise removal, next priority level contain a contrast enhancement and correction of object borders. Priorities determine the order of the functions and need for additional analysis using neural network, for example.

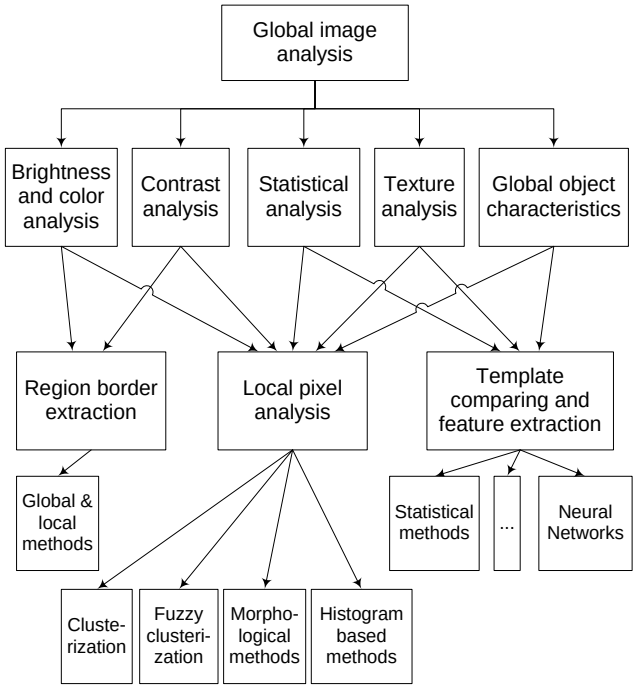

Fig. 2. Selection of functions for image processing based on image features.

Image processing functions relate to basic computer vision topics. This is corresponding to their application for image changing. Every function changes properties of an image and is appled for specific processing cases. Every function is described in an interpreter table and can be supported by additional information.

B. Software development for histological image analysis

For elaboration of a structural scheme of a software basis interface an estimation of functionality and compatibility of existed software development tools were done. Tasks which may be solved with software to be developed, initial data, diagnostic features and characteristics have been observed.

Based on the material posted by Guillaume Marceau [1], who in his study used parameters of 72 implementations of programming languages, and compared them to 19 special tests, prepared by the project "The Computer Language Benchmarks Game" as a kernel chosen interpreter Lua [6].

In the first case, data are processed using temporary file, which allows to analyze records (images). In another case, a transfer by calling a run-time library is performed (Fig. 3).

Our system is based on the interpreter of Lua language. It was elaborated as the main module, which may provides an interaction of complex components. It includes a graphical interface, global variables, and image structure. Architecture of graphical interface has been carrying out by linking the Highgui library [7] from OpenCV package [7]. The image processing and analysis functions are supported by OpenCV library, but connection Lua with OpenCV are realized by Lua-binding interface for connection function of OpenCV with Lua.

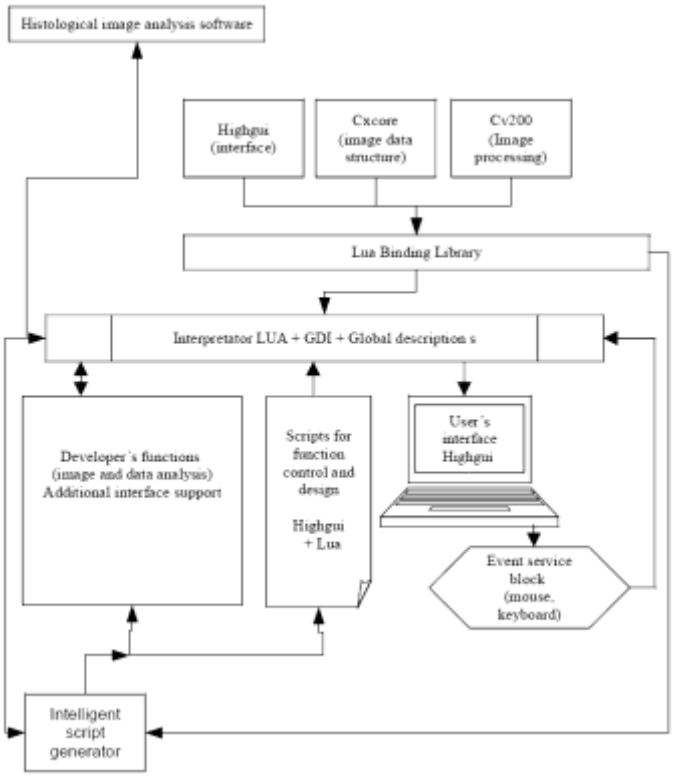

Fig. 3. A variant of our bsic softare components for intelligent image processing.

An image structure is determined by a module of graphical interface into OpenCV library, which is responsible for visualization and representation of images. Headers of image structures are global variables-pointer of interpreter Lua and have special type - userdata. Userdata correspond to pointers in computer address space. This module also includes the image read/write function, simple functions of image processing and interactive contouring. All interactive functions return values in the event block, which changes global variables of the interpreter. For tasks of monitoring space-occupying lesion a simultaneous usage of several modules are required. In this case an interaction has been performing by using global variables of the Lua interpreter and properties of userdata type of the interpreter.

During image processing, a node of Lua can sends images through Lua-outlets. It should first send images with the highest Lua-outlet identificator. During the processing commands in scripts may be changed and this affects the outcome. If the program is changed by lua-script the new sequences of command change method. This methods registers this sequences of command as wanting observer notifications. It is depended from new properties of changed image. At the end of script processing loop notifications are sent eventually from a slow thread. Going through each of the nodes registered as needing observer notification, the notify event-slot is called. This slot calls a node state method to get the dictionary with the node attributes and sends the attributes to the observers.

Communication between nodes in the same script is done through direct function calls by event-slots, either using the virtual method bang for the first inlet or a function pointer for other inlets. Images are processed by reference.

Adding new complex functions for processing and analysis is carrying out by a group of developer's (simple) functions. As an input new functions may get any global variable or text and numerical constants from the Lua interpreter.

All internal controlling of software is carrying out by text scripts of Lua, which are divided into two categories:

1. Scripts for image sequences analysis;

2. Scripts for operative functionality and setting up of software at workplace of medics.

All scripts are stored in a text format and easily accessible. However they cannot be edited by users, but may be edited by developers only.

Scripts manage to software organization and create new additional functions for image analysis and processing. The module of script generation defines new functions as sets of simple functions from the libraries described above. It includes intelligent components for connection results of image processing and image characteristics. Of course such variant can be realized only for particular tasks, in our case the tasks oriented for histological image analysis.

Every interpreter defines functions through a specific table. We use it for definition of connection image characteristics with a function in our software. In result such software has intelligent self-programming possibilities.

A script generation module consists of two parts: image analysis and script construction. The fist part starts from global image analysis that include histogram analysis and basic

statistical analysis for pixels distributions, fractal and texture analysis. On the base of such analysis estimations of noise, blurring, and some image characteristics are calculated. From the interpreter function table an image preprocessing script is generated for image enhancement.

Than local image analysis is applied by convolution with different filters and statistical analysis of line-profile characteristics. This analysis allows to estimate characteristic of cell borders and contrast. Such estimation defines functions for image contrasting and border underlining. After generation quality of the outcome image is tested. If the image quality is low the process of image analysis and function definition should be repeated. Such procedure generates image improving scripts that can be change by a user or developer. This script is only a proposition and need in a user control. The same mechanism of image analysis and function definition works for stages of image segmentation and post-processing.

In result, the software generates a script that corresponds to a set of functions for an object extraction. It can be used for extraction of histological objects in an image. We scripts for nuclear extraction from histological images. Then characteristics of objects are calculated. It is necessary to detect type of objects that are presented in the image. We divide objects for five basic types: blobs, front, needles, dendrites and nets. Such procedure of object detection are spending by function through script generation. Using global fractal as texture characteristics software detect the geometrical type of objects and forms characteristics sets for an object description.

For definition an image processing function in scripts we try to use the Kohonen neural network (self-organizing map or SOM) [8, 9]. It is a class of neural networks, the main element of which is a Kohonen layer. The Kohonen layer consists of adaptive linear combiners. Estimations of global conditions of an image are used as weights in the neural network. Adjusting of the input weights and vector signal quantization is closely related to a simple basic algorithm for cluster analysis (for example, the method of dynamic cores or K-means).

Our system is supported by a script generator module. The module uses a Lua-metatable of functions for image processing and a corresponding table with estimations of images. Such a table definition of function sets are realised by Kohonen neural network. In result the module proposed scripts for different tasks of image processing as text files. Users can spend analysis of these scripts and change something in them.

III. CONCLUSION

In this paper, we have classified histological objects by their topological properties, which allow to determine a set of histological characteristics in terms of image processing functions. We propose a scheme of automatic generation of scripts describing sets of simple image processing functions for different tasks of analysis of histological tissues and we describe software for this generation. This software is based on principles of open architecture and allows to change design and possibilities of it in real time on physician work place without compilation stage. In other side the program run time remains the same as in a compiled version. The suggested software architecture simplifies program development for analysis of medical images, in particular histological ones.

ACKNOWLEDGMENT

This work is supported by ISTC projects #B-1682 and #B-1636.

REFERENCES

[1] G. Marceau “The speed, size and dependability of programming languages,” in *Blog “Square root of x divided by zero”, Saturday, May 30, 2009* (http://gmarceau.qc.ca/blog/2009/05/speed-size-and-dependability-of.html).

[2] G. Reitmayr, D. Schmalstieg. “An open software architecture for virtual reality interaction,” *Proc of VRST'01*, November 15-17, 2001, Banff, Alberta, Canada, pp.47-54.

[3] A. Nedzved, A. Belotserkovsky, T.M. Lehmann, S. Ablameyko “Morphometrical feature extraction on color histological images for oncological diagnostics,” *Proc of 5th International Conference on Biomedical Engineering*, 14-16 February, 2007, Innsbruck, pp.379-384.

[4] T.Kanade, Z.Z Yin,., R. Bise, S. Huh, S. Eom, M.F. Sandbothe, M. Chen “Cell image analysis: Algorithms, system and applications,” *Proc. of WACV11*, 2011, pp.374-381.

[5] L. He, L. R. Long, S. Antani, and G. Thoma, “Computer assisted diagnosis in histopathology,” Z. Zhao ed., *Sequence and Genome Analysis: Methods and Applications*, ch. 11, iConcept Press, 2010. (http://www.iconceptpress.com/books/publicationContent.php?publication id=B00003 Accessed 11 November 2010).

[6] R. Ierusalimschy, *Programming in Lua*, 2nd ed., Lua.org, March 2006 ISBN 85-903798-2-5.

[7] G. Bradski, A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*. O’Reilly, 2008.

[8] S.. Kaski, “Data exploration using self-organizing maps,” *Acta Polytechnica Scandinavica, Mathematics*, Computing and Management in Engineering Series No. 82, Espoo 1997, 57 pp.

[9] A Ultsch “U*-Matrix: a tool to visualize clusters in high dimentional data,” University of Marburg, Department of Computer Science, Technical Report Nr. 36: 2003. pp.1-12.