DRAFT: A.Nedzvedz, A.Belotserkovsky, I.Gurevich, Yu.Trusova, V.Yashina. Self-Developing Software for Image Processing Tasks // 8th Open German-Russian Workshop "Pattern Recognition and Image Understanding" - OGRW-8-2011, November 21-26, 2011, Nizhny Novgorod, The Russian Federation: Workshop Proceedings. - Nizhny Novgorod Lobachevsky State University, 2011. - P.215-218.

SELF-DEVELOPING SOFTWARE FOR IMAGE PROCESSING TASKS

A.Nedzvedz*, A.Belotserkovsky*, I.Gurevich, Yu.Trusova**, V.Yashina****

***United Institute of Informatics Problems, National Academy of Sciences of Belarus, 6, Surganova str., 220012 Minsk, Republic of Belarus, {NedzvedA, abelotser}@newman.bas-net.by**
**** Institution of the Russian Academy of Sciences "Dorodnicyn Computing Centre of RAS", 40, Vavilov str., 199333 Moscow, the Russian Federation, {igourevi, ytrusova}@ccas.ru, werayashina@gmail.com**

A new methodology of the development of intelligent software for medical image analysis is proposed. The methodology is based on combining an intelligent agent and an interpreter. It works as a script generator for creating new sequences of image processing functions. The script generator forms new image processing scripts after analysis of image properties.

Introduction

Computer engineering includes many different parts that influence to properties of software. Usually, the supporting of software tools has specific requirements to software organization. It depends on solving tasks, the software user skills, and the environment of user's workplace. Therefore, modern requirements to image analysis software are dynamical organization of functionality, user interface, data management for image processing methods, quality and distance control, etc.

Any software application in a given domain deals with the study of object in terms of their origin, structure, and functional properties (for instance, medical tasks [1]). The practical implications of such study are of key importance in scientific field since they consist of: 1) recognition of object types, fundamental for classification; 2) analysis of object interactions and reactions, useful in recognition problems; 3) detection of scenes to generate the overall picture of interest; 4) practical context of scenes in certain application domain.

It is not a secret that though human expertise it is hard, if not impossible (at least at the moment), to replicate in computerized applications automated image analysis tools if we need a full universal support of workplace in several ways. Actually, current advanced study can easily obtain high-resolution images from which a considerable number of relevant measurements (i.e., features) can be extracted and used for significant investigations. The automation of feature extraction process yields an objective, quantitative, detailed, and reproducible computation of morphofunctional characteristics and allows the analysis of large quantity of images.

In the following sections, the domain of software for image analysis is summarized and then the proposed suite of ontologies is illustrated in detail. The last section describes a scheme of experimental intelligent tool developed for real image analysis problems.

Software properties

Any software may be divided into compiled programs and interpreters. Sometimes the 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 is usually used as stored history of image processing operations for applying to new image. In this case, the interpreter allows us to create additional simple functions on a base of kernel possibilities without changing the software itself [1]. We propose to use the interpreter as a kernel for our software. In this case, the interpreter is used as a manager of action. The kernel provides functions execution, calculations, and supporting of the GUI events.

Our software is based on a mixture of a compiled library and an interpreter with image

DRAFT: A.Nedzvedz, A.Belotserkovsky, I.Gurevich, Yu.Trusova, V.Yashina. Self-Developing Software for Image Processing Tasks // 8th Open German-Russian Workshop "Pattern Recognition and Image Understanding" - OGRW-8-2011, November 21-26, 2011, Nizhny Novgorod, The Russian Federation: Workshop Proceedings. - Nizhny Novgorod Lobachevsky State University, 2011. - P.215-218.

processing functions. The software is intended for professional software developers and software designers as well as for all users who need to process images for solving their tasks. The interpreter has possibilities for including additional functions from a compiled dynamical library. It allows us to change software properties and software applications without compiling stage. On other side, users can modify the graphical interface to improve the efficiency of their work. We apply such technique for development of image analysis software for medicine, satellite technology and quality control [3].

In addition, the new workplace is changing the software requirements. It is necessary to estimate functionality and compatibility of existing software development tools to define dynamically changing properties of the development system of image processing software.

Furthermore, the ontology is equipped with a library of algorithms for feature computation. This way, it can have a number of significant, applicative perspectives: once integrated into specific automated image analysis tools, it can be used for [7, 8]:

• aiding image analysis, by making the results quantitative, reproducible, and objective;
• selecting relevant subsets of features in diagnostic tasks, by tracing the set of features most commonly used by experts;
• discovering new knowledge, by allowing the extraction of novel features and the execution of statistical tests for evaluating their significance;
• supporting decision making since it can be considered as a first step in the development of a knowledge base for automated analysis and diagnosis.

Features of image analysis

In this section, basic features of image analysis that can be used for determination of a way of processing are described.

A. Common processing sequence for histological samples image analysis

Usually, processing of histological images can 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. The application of a function depends on initial data or image estimations. It is possible to define such estimations in many cases, for example, for contrast, noise or blurring. We can construct table of image processing functions and image estimations.

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

Based on initial image characteristics, it is possible to define a function for image processing and analysis (see Fig. 1).

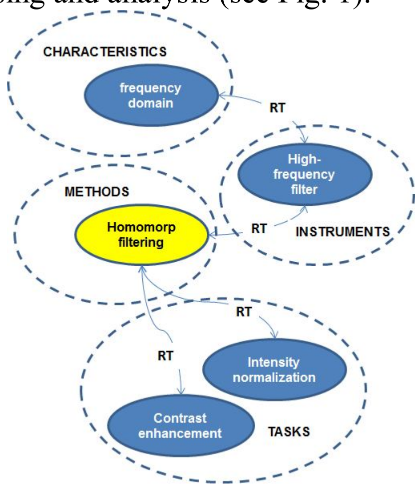

Fig 1. Scheme of homomorphic filtering definition based on image characteristics.

As a result, the table of connection function and image estimation are constructed.

In each step, functions are indicated by priorities. For example, for the image improvement step the higher priority is defined for noise removal, next priority level includes contrast enhancement and correction of the borders. Priorities determine the order of the functions application and the need for re-analysis using neural network.

Image processing functions relate to basic computer vision topics. Every function changes properties of image and is applied for specific processing cases. Every function is introduced in an interpreter table and can be supported by additional information.

DRAFT: A.Nedzvedz, A.Belotserkovsky, I.Gurevich, Yu.Trusova, V.Yashina. Self-Developing Software for Image Processing Tasks // 8th Open German-Russian Workshop "Pattern Recognition and Image Understanding" - OGRW-8-2011, November 21-26, 2011, Nizhny Novgorod, The Russian Federation: Workshop Proceedings. - Nizhny Novgorod Lobachevsky State University, 2011. - P.215-218.

Development tools for image analysis

This section describes basics of a flexible suite of developed software tools for medical image analysis. Tasks that may be solved with the proposed software as well as initial data, diagnostic features and image characteristics have been observed. We use C-language from Microsoft Visual Studio for software development.

At first, data are processed using temporary file, which allows us to analyze records (images). Then, a transfer by calling a run-time library is performed (Fig. 2).

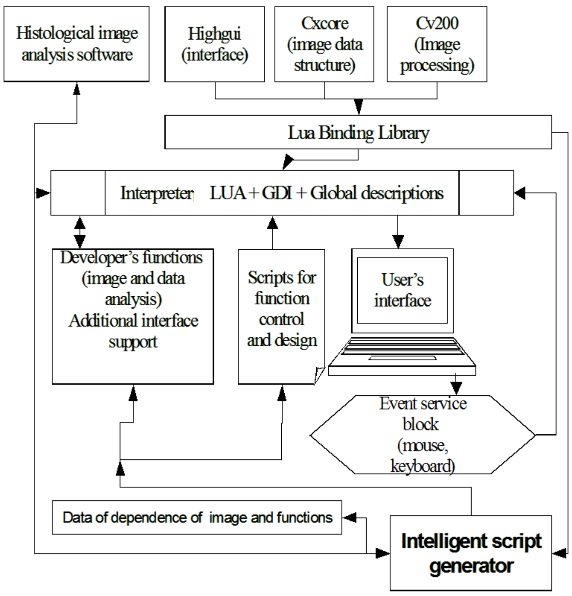

Fig. 2. Scheme of segmentation methods definition based on image characteristics.

The system is based on the interpreter of Lua language and was elaborated as the main module, which may provide interaction of complex components.

Lua is a register-based virtual machine. Traditionally, most virtual machines intended for actual execution are stack-based ones.

Algorithms under Lua are optimized by tables and used as arrays: unlike other scripting languages, Lua does not offer an array type. Instead, Lua programmers use regular tables with integer indices to implement arrays. Lua uses a new algorithm that detects whether tables are being used as arrays. These algorithms automatically store the values associated to numeric indices in an actual array, instead of adding them to the hash table (see Fig.3).

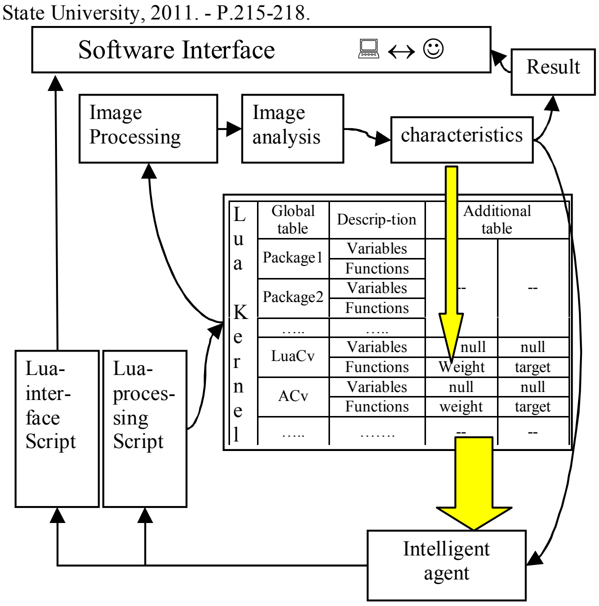

Fig. 3. Script generation cycle for image processing and analysis.

Lua supports first-class functions with lexical scoping.

This mechanism uses an array-based stack to store activating records. Lua uses a novel approach to function closures that keeps local variables in the array-based stack and pushes them to the heap.

Every package included in Lua is represented in such table-array. We set two additional columns in this table to define vector of probability coefficient of image characteristics and vector of coefficient of the target. In this case the target is considered as characteristics of the image processed by current function. These vectors allow to select image processing function on a base of characteristics analysis.

All vectors are collected by comparing with image characteristics by means of Lua-table. This collection is sending to intelligent agent to generate a new script. Script processes the image and changes characteristics. These characteristics are pushed to Lua-table then to generate a new collection for intelligent agents and to represent in user interface together with processed image. In result software modify itself and changes processing properties and its interface.

Such interpreter includes a graphical interface, global variables and image structure. The architecture of the graphical interface is supported by linking Highgui library [7] from OpenCV package [7]. Functions for image processing and analysis are supported by

OpenCV library, but linking Lua with OpenCV is realized by Lua-binding interface. 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 – user data. User data correspond to pointer in computer address space. This module also includes 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 interpreter. For tasks of monitoring space-occupying lesion a simultaneous usage of several modules is required. In this case, an interaction has been performing by using global variables of the LUA interpreter and properties of user data type of Lua interpreter.

All internal control of the software is supported by text scripts of LUA, that are divided into two categories: 1) scripts for image sequences analysis; 2) scripts for operative functionality and setting up of software at workplace of physicians.

All scripts are stored in a text form and easily accessible. However they are not for changing by user and may be edited by developers only.

Scripts manage software organization and create new additional function for image analysis and processing. The analysis of image pre-processing allows us to define processing functions. The module of scripts generation defines such sets. It includes intelligent components for connection of results of image processing functions and image characteristics.

Conclusion

In this paper, we proposed the scheme for automated generation of sets of image processing functions for image analysis and described the supporting software. This software is based on principles of open architecture and allows us to change its design and possibilities in real time at workplace without compilation stage. In other side, the speed of the program remains the same as in the compiled version. We assume that it is necessary to change intelligent agent for script generator. But here we described suitable way to develop image processing software. Main feature of such software is an ability to dynamically modify the user interface and the way of image processing on a base of image characteristics.

Acknowledgements

This work was partially supported by the Russian Foundation for Basic Research (project nos. 10-01-90013), by the Program of the Presidium of the RAS "Intelligent information technologies, mathematical modelling, system analysis and automation" (project 204) and by ISTC project #B-1682 and #B-1636.

References

1. G.Marceau "The speed, size and dependability of programming languages", Blog "Square root of x divided by zero", 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.
3. 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
4. A.Nedzved, A.Belotserkovsky, T.M.Lehmann, S. Ablameyko "Morphometrical Feature Extraction On Color Histological Images For Oncological Diagnostics", 5th International Conference on Biomedical Engineering, 14-16 February, 2007, Innsbruck: Proc. - P.379-384.
5. R.Ierusalimschy "Programming in Lua", (second edition) Lua.org, March 2006 ISBN 85-903798-2-5
6. G.Bradski, A.Kaehler "Learning OpenCV: Computer Vision with the OpenCV Library", O'Reilly, 2008.
7. S.Colantonio, I.Gurevich, G.Pieri, O.Salvetti, Yu.Trusova. Ontology-Based Framework to Image Mining // Image Mining Theory and Applications: Proceedings of the 2nd International Workshop on Image Mining Theory and Applications - IMTA 2009 (in conjunction with VISIGRAPP 2009), Lisboa, Portugal, February 2009 / Edited by I.Gurevich, H.Niemann and O.Salvetti. – Portugal: INSTICC PRESS, 2009. – P.11-19.
8. I.B.Gurevich, O.Salvetti, Yu.O.Trusova. Fundamental Concepts and Elements of Image Analysis Ontology // Pattern Recognition and Image Analysis: Advances in Mathematical Theory and Applications. - Pleiades Publishing, Ltd., 2009. - Vol.19, No.4. – P.603-611.