UNITED INSTITUTE OF INFORMATICS PROBLEMS
OF THE NATIONAL ACADEMY OF SCIENCES OF BELARUS

Pattern Recognition and Information Processing

PRIP'2014

PROCEEDINGS OF THE 12th
INTERNATIONAL CONFERENCE
28–30 May 2014, Minsk, Belarus

Minsk
UIIP NASB
2014

UDC 004.93'1;004.932

Pattern Recognition and Information Processing (PRIP'2014) : Proceedings of the 12th International Conference (28–30 May 2014, Minsk, Belarus). – Minsk : UIIP NASB, 2014. – 340 p. – ISBN 978-985-6744-81-8.

This Proceedings book contains papers accepted for publication and presentation at the 12th International Conference "Pattern Recognition and Information Processing" that will be held in May 28–30, 2014, Minsk, Belarus.

Proceedings are prepared for publication by PRIP'2014 Program Committee and Belarusian Association for Image Analysis and Recognition. They could be useful for students and researchers working in the following areas: Pattern recognition and Image analysis, Knowledge engineering, Big data, Decision support systems, Fuzzy mathematics and Fuzzy systems, as well as Applications of pattern recognition and image analysis methods.

Papers are published in the form presented by authors who are fully responsible for correctness of the reported results and technical quality.

Сборник включает доклады 12-й Международной конференции "Распознавание образов и обработка информации", которая состоится 28–30 мая 2014 г. в Минске (Беларусь). Сборник подготовлен к выпуску программным комитетом конференции и Белорусской ассоциацией по анализу и распознаванию изображений.

Доклады конференции будут полезны студентам и специалистам, работающим в следующих областях: распознавание образов и анализ изображений, обработка и представление знаний, "большие" данные, системы поддержки принятия решений, основанные на знаниях, нечеткая математика и системы, а также практические приложения методов распознавания образов и анализа изображений.

Статьи печатаются в виде, представленном авторами, которые несут всю ответственность за достоверность приводимых научных результатов.

Editors:

Alexander Tuzikov,
Vassili Kovalev

ISBN 978-985-6744-81-8

Organized by

United Institute of Informatics Problems of the National Academy of Sciences of Belarus

SYSTEM FOR AUTOMATIC PARALLELISATION OF PROCESSING LARGE VOLUMES OF IMAGES USING MULTI-USER RESOURCES

A. Nedzved[1], A. Belotserkovsky[1], A. Khmelkov[2]
[1]United Institute of Informatics Problems of the NAS of Belarus, Minsk;
[2]Belarusian State University, Minsk
e-mail: nedzveda@tut.by

In this paper, we considered a problem of specific software development for parallel processing of large volume of image. It is based on concept of generating a list of tasks and distributed at different computational resources. The effectiveness of such a system is gained due to interaction between database, command interpreter for image analysis and monolithically compiled programs.

Introduction

It is quite complicated to replicate in computerized applications tools for automated image analysis if we need a full universal support of workplace in several ways [1]. Thus, we come to the necessity of using the already popular notion of Software Flexibility. It mustn't be confused with a flexible development methodology – "Agile software development", – the concept, which is quite popular as a new feature of software applications. The generally accepted definition of software flexibility has been given by IEEE [2] (Flexibility: "The ease with which a system or component can be modified for use in applications or environments other than those for which it was specifically designed"). It can be challenged on a number of positions. In addition there is already a number of metrics at the moment, characterizing the degree of flexibility of the developed environment [3] but sometimes it contradict with each other. This issue is actually not the subject of this paper, since authors are pursuing the global goal to offer the architecture of applied task-driven system that would later allow:

– 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.

At this stage, we propose conceptual schemes for process management of image processing system (as an application domain), and generating a user interface, with the intellectual, social (collaborative, recommending) component according to specific needs of certain users and developers. It allows modifications and building-up the functionality due to the wide use of workflows (processing scripts), which obviously will contribute to solve a problems described above.

1. Properties of Image Analysis Software

As of applied needs there are plenty of modern software for image analysis already developed at the moment. It sense of way they were built we can distinguish six types of applications:

– application programming interface (**API**) – a set of functions or routines that are designed to solve a specific task or are allowed to interact with a specific software component (for example VIGRA, IM);

– **script application** – number of instructions (a programm) writted using scriprt paradigm for special run-time environment that can interpret (rather than compile) and automate the execution of tasks which could alternatively be executed one-by-one (for example luaCV scripts , TCL-ITK);

– **plug-in** – a software component that adds a specific feature to an existing software application (for example ImageJ);

– software **component** – is a software package, a web service, a web resource, or a module that encapsulates a set of related functions (or data) (for example toolboxes for Matlab, Matcad);

– **event component** are handled synchronously with the program flow, that is, the program has one or more dedicated places where events are handled, frequently an event loop (for example IMAQ);

– **source code** is any collection of computer instructions (possibly with comments) written using some human-readable computer language (OpenCV, ITK).

Though these types are mixed for different software packages (for instance, libraries IM, OpenCV may b classifed as API and source code also), each type of software has certain differences and properties. They either offer a high degree of *adaptability* with little or no *assurance over consistency* or severely restrict change to achieve consistency (fig. 1). Source code, depicted in the lower right corner, offers complete adaptability because it permits third parties to change the applications source code, its principal behavioral specification. In theory, any change is possible. In practice, the burdens of understanding a source code good enough to make changes coupled with verifying the correctness of those changes poses significant hurdles, especially for large intricate software systems. While any change is possible, few if any assurances are provided over those changes. In response, various strategies have been devised that help raise confidence in source code changes, and hence an applications consistency. These include confining change to certain source files and verifying application invariants using embedded assertion statements or test scripts. The consistency gained using these measures cuts off certain avenues of adaptability.

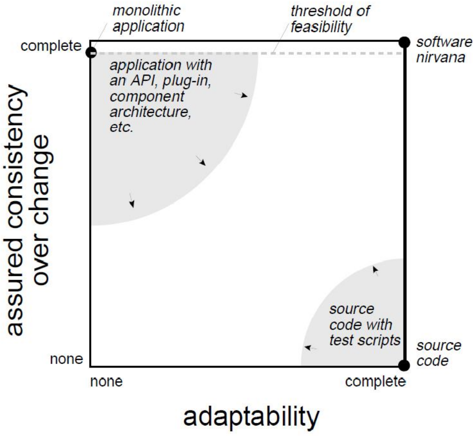
monolithic application
threshold of feasibility
complete
software nirvana
application with an API, plug-in, component architecture, etc.
assured consistency over change
source code with test scripts
none
source code
none
complete
adaptability

Fig. 1. Dependency between adaptability and consistency offered by current realization techniques for image analysis software

Monolithic applications, which lack such mechanisms, represent the opposite extreme. By prohibiting third party change, application developers can guarantee a high degree of consistency. Incorporating techniques of developing into these applications, by utilizing APIs or component architectures for example, increases adaptability by permitting third parties to change certain aspects of the application, but simultaneously introduces the possibility that those changes could violate consistency.

2. Tools for Image Analysis

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. The system was designed as the main module. It provides interaction of complex components as it is built on the interpreter of Lua language

Lua is a register-based virtual machine. Traditionally, most virtual machines intended for actual execution are stack-based ones. Algorithms, designed with Lua, are optimized by tables and used as arrays: unlike other scripting languages, Lua does not offer an array type. Instead of it 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 (fig. 2). This mechanism uses an array-based stack to store activating records. There is 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.

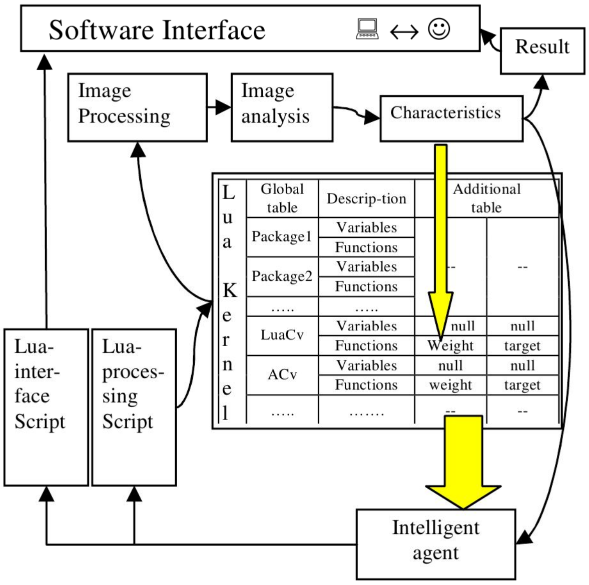

Fig. 2. Script generation cycle for image processing and analysis

All vectors are collected by comparing with image characteristics by means of Lua-table. This collection is sending then 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 modifies itself and changes processing properties and 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.

3. Multitasking system for big data image processing tools

A principle scheme to process and analyze large image volumes based on task tables is proposed was proposed were each of computer systems has its own priority for any task . At every step, functions are indicated by priorities. For instance, for image enhancement the higher priority is defined to noise removing, next priority level includes contrast enhancement and correction of the borders. Priorities determine the order of functions application and necessity of re-analysis using Neural Network. The variable of priority determines location of function in the generated script. In consequence such software has intelligent self-programming possibilities. the concept is based on combination of dynamic libraries, databases and the interpreter features with a set of image processing functions. As a result, the software can be divided into two parts: the first one is focused on an independent application design, and the second one is related to applied users. The interpreter has an opportunity to use additional functions of dynamic libraries. It allows to change software properties and avoide the compilation stage. On the other hand, users can change a task in a database, which allows to parallelize and control its execution process.

It is known that idea of parallel computing is based on the fact that the problems can often be divided into a set of smaller problems, which can be solved simultaneously. Parallel computing is presented in several forms: the bit level parallelism, instruction level parallelism, data parallelism, and task parallelism. In our case, data parallelism will be used. The main idea of the approach based on the data parallelism is that of one operation being performed on all the data array elements. Various fragments of such an array are processed using different processors being nearby or distributed in a computer network. The data distribution between the processors is carried out by the software.

In common, system is constructed from four basic software components: file server for images set, data base for tasks management, manager for generating sets of tasks and manager for task execution (fig. 3).

File server stores all images (original one and processed). All managers are realized as client application. The manager of tasks generating spends estimation of images from file server. Evaluation of image marks is performed based on a specialized tool as described above This manager creates a list of tasks by linkages image estimation marks with commands of image analysis and processing solutions and user decision. This list is stored in the database. Simple manager of tasks execution takes first command with special status form database and push it to execution. It changes status of command for every stage of processing. In result sets of images are processed by different computing resources at the same time. It decreases a computational cost of image analysis. Parallel computing systems are physical computing and the software systems performing, in one way or another, parallel data processing on many computational nodes. On each processor of the multiprocessor system, a single-threaded process is implemented which communicates with the processes running on other processors using the messages. One sequential function that processes one image (or group of the coupled images) is developed, and the system provides application of this function to large image volume in a parallel mode. It allows to focus on developing the image processing algorithms while avoiding the distraction related to parallel processing implementation.

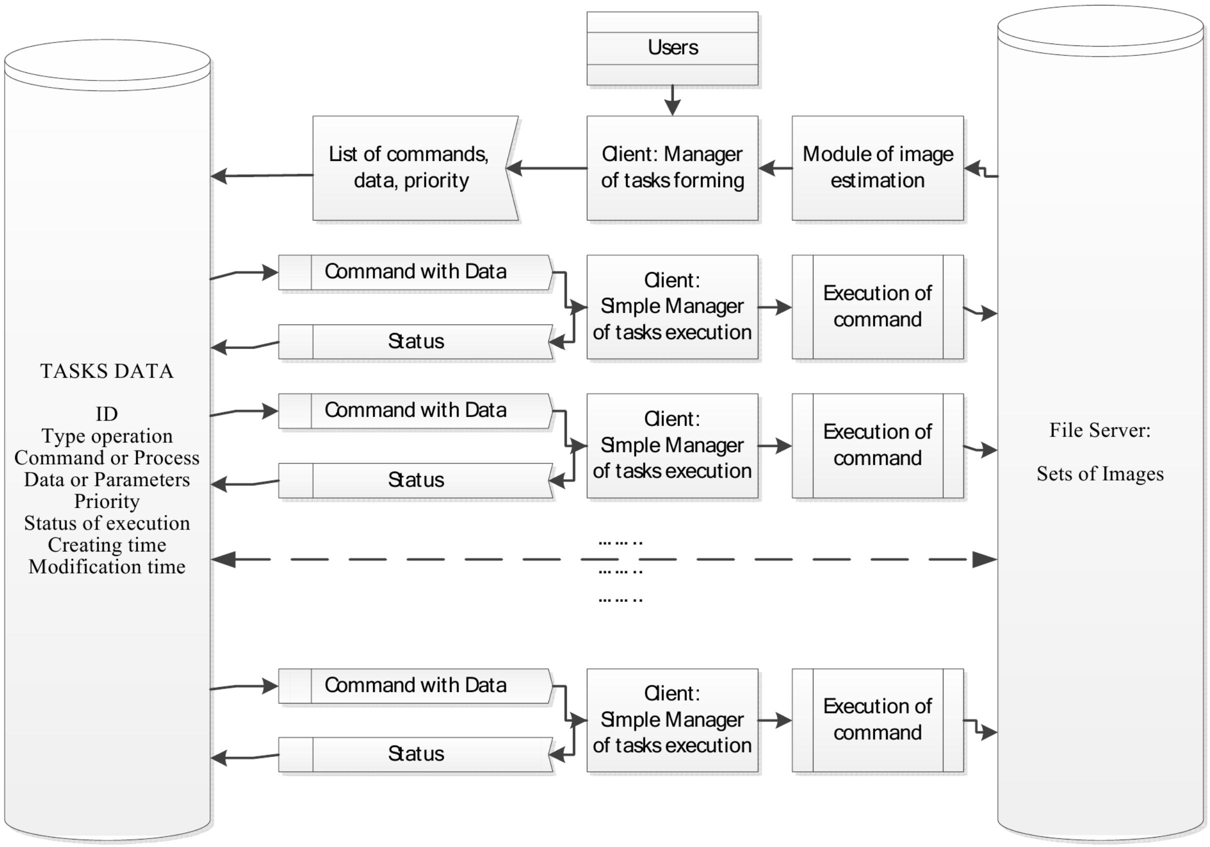

Fig. 3. Information flows of multitasking system for big data image analysis

Conclusion

The technology of software development based on the automatic generation of the scripts including the set of simple image processing functions for a variety of problems of large image sequences analysis. This software is based on the open architecture principles and allows to change the design on the user workstation without compilation stage on the real-time basis. At the same time, the program execution time remains the same as the one in the compiled version. The software dynamism is significantly improved as using the interpreter components, the software can be easily modified and adapted to solving new problems. In addition, the overall concept of such software development will definitely show its worth in complex software systems designing in the nearest future.

References

1. Self-Developing Software for Image Processing Tasks / A. Nedzvedz [et al.] // Proc. on Open German-Russian Workshop “Pattern Recognition and Image Understanding”. – Nizhny Novgorod, 2011. – P. 215–218.

2. IEEE. Standard Glossary of Software Engineering Terminology 610.12-1990. – Los Alamitos : IEEE Press, 1999. – Vol. 1.

3. Eden, A.H. Measuring Software Flexibility / A.H. Eden, T. Mens // IEE Software. – London, UK : The Institution of Engineering and Technology, 2006. – Vol. 153, no. 3. – P. 113–126.

4. Oreizy, P. Open Architecture Software: A Flexible Approach to Decentralized Software Evolution / P. Oreizy. – Irvine : University of California, 2000. – 142 p.

5. Gurevich, I.B. Fundamental Concepts and Elements of Image Analysis Ontology / I.B. Gurevich, O. Salvetti, Y. Trusova // Pattern Recognition and Image Analysis: Advances in Mathematical Theory and Applications. – Pleiades Publishing, Ltd., 2009. – Vol. 19, no. 4. – P. 603–611.

6. Bradski, G. Learning OpenCV: Computer Vision with the OpenCV Library / G. Bradski, A. Kaehler. – O'Reilly, 2008.

7. Ierusalimschy, R. Programming in Lua / R. Ierusalimschy. – 2nd ed. – Lua.org, 2006.

CONTENTS

Abaturov V., Dorogov Yu. The knowledge extraction analytical platform for embedded industrial applications........ 6

Abramovich M., Mitskevich M. Spatio-temporal cluster analysis of disease........ 11

Alexandrov O., Svirsky D., Radoman N. Modeling of flow distribution in electric network taking into account the uncertain factors........ 16

Batmunkh Amar State space representation of multi-input multi-output circuits........ 21

Bat-Erdene B., Ganbat Ts. Computer model for distinguishing Asian people by face........ 27

Bobkov A., Gafurov S., Krasnoproshin V., Romanchik V., Vissia H. Information extraction based on semantic patterns........ 30

Britenkov A., Dokuchaev A. Structure-functional approach to modulating function trend estimation based on time-series data under uncertain conditions........ 35

Dimitrakopoulos G. Reconfiguring driving styles based on bayesian networking principles........ 40

Dorogov A.Yu. Fractal learning of fast orthogonal two-dimensional transformations........ 45

Dvoenko S. Meanless *k*-means as *k*-meanless clustering with the bi-partial approach........ 50

Evsutin O., Shelupanov A. Creating of the discrete orthogonal transformations with the use of cellular automata dynamics........ 55

Faizullin R.T., Ogorodnikov Y.Y. Recognition of zero bits of 3-sat problem by applying linear algebra's methods........ 60

Ganchenko V., Doudkin A., Petrovsky A. Decision support for precision farming complex........ 65

Gladun A., Rogushina J., Andrushevich A., Kurbatski A. User-oriented recognition of intelligent information objects in distributed dynamic informational web-space........ 70

Hancharou D., Nedzved A., Ablameyko S. Skeletonization algorithm of high resolution vascular data........ 76

Hetsevich Yu., Mandzik V., Rusak V., Hiuntar A., Okrut T., Lobanov B., Lysy S., Dzenisiuk Dz. The system of generation of phonetic transcriptions for input electronic texts in Belarusian........ 81

Kanonik D. Distinguishing mental activity types by electroencephalography records........ 86
Kazlouski A., Sadykhov R. Plain objects detection in image based on a contour tracing algorithm in binary images........ 89

Karkanitsa H. Development of domain-specific language for constructing dynamic subject domains........ 94

Kharin Yu., Zhurak M. Statistical analysis of spatio-temporal data by poisson conditional autoregressive model 98

Kharinov M. Hierarchical pixel clustering for image segmentation 103

Khorev I.E., Zelepugin S.A., Tolkachev V.F., Zelepugin A.S. Image analysis in high-velocity interaction of a group of bodies with a target 108

Klionskiy D.M., Kaplun D.I., Voznesenskiy A.S., Gulvanskiy V.V. Study of digital filter banks and their software-hardware implementations for wideband monitoring 114

Kliuchenia V., Petrovsky A. Architecture of the DCT-IDCT processor for lossless scheme coding 119

Kovalev A. An agent-based model of German-French inter-border residential mobility in Strasbourg region 125

Krasnoproshin V., Mazouka D. High-level rendering pipeline construction tools 129

Krasnoproshin V., Obraztsov V., Vissia H., Nguyen Q.T. Data mining and knowledge-based technology 135

Kuzmich A.I. Standartization of mobile heterogenious objects monitoring based on pattern recognition 140

Lazebnikov S., Boudnik A., Sarychev A. Application of big data technologies to image processing 145

Liauchuk V.A., Snezhko E.V. A method for coarse segmentation of pathological lungs on CT images using ribcage 152

Lisitsa Y., Yatskou M., Apanasovich V., Apanasovich T. Studying the advanced segmentation methods with the computer-simulated images 157

Lisitsa Y., Yatskou M., Skakun V., Digris A., Shingaryov I., Apanasovich V. Simulation model to study denoising methods 162

Losik G., Brazevich H. Brain coding of information on the object shape by its physical model 168

Makovetskii A., Kober V. Frequency analysis of gradient descent metod for imagerestoration 171

Mamaev N., Lukin A., Yurin D. HENLM-3D: A 3D computer tomography image denoising algorithm 176

Mashtalir S., Mikhnova O. Video content analysis and key frame extraction method 181

Miramontes-Jaramillo D., Kober V., Karnaukhov V. Fast image matching with sliding hogs 187

Moghaddam R.A., Mortazavi S.M., Malekzadeh M. Reducing energy consumption in wireless sensor networks by CICA algorithm in comparison with the genetic algorithm 191

Murashov D. Application of structure tensor features for describing an artistic manner 196

Nechval N.A., Nechval K.N., Bausova I., Berzins G., Krasts J., Strelchonok V.F., Moldovan M. Distance-based embedding procedures for pattern classification 201

Nechval N.A., Nechval K.N., Bausova I., Berzins G., Krasts J., Strelchonok V.F., Moldovan M. Statistical validation of simulation models 207

Nedzved A., Belotserkovsky A., Khmelkov A. System for automatic parallelisation of processing large volumes of images using multi-user resources 213

Nedzvedz O., Ablameyko S. Detection patterns of living cells from its aggregation on the digital image........218

Noskov M., Tutatchikov V., Kolcova I. Three-dimensional Fast Fourier Transform algorithm modification by analogue of Cooley-Tukey 223

Orlov A.A., Astafiev A.V., Popov D.P. Development of algorithm for localization of symbolic with the use of analysis of the color map 227

Paringer R.A., Kupriyanov A.V. Research methods for classification of the crystallograms images........231

Polyakova G. Logical-and-probability simulation model of data processing and analysis........235

Pshenichny D., Dvoenko S. Elimination of the non-positive definiteness of matrices of pairwise comparisons........240

Semenov M., Semenov D. Targets visualization tool to explore testing results 244

Sergeev R.S., Tuzikov A.V., Kavaliou I.S. Mutation analysis of genetic sequences and building mutational database in context of tuberculosis treatment 249

Shangytbayeva G. Planning of the enterprise resources........254

Shestov A., Kumskov M. Multiscale molecular surface descriptors........257

Shut O. The complexity of inductive resolution algorithms in pattern recognition problems........260

Sidorenko A.V., Shakinko I.V. The modified principal component analysis of information encrypted using deterministic chaos 265

Snezhko E., Kharuzhyk S., Tuzikov A., Kovalev V., Safonau I. A scheme for nodules detection on CT images of lungs........270

Starovoitov V. Image processing for fundus image analysis........275

Sychyou U. Method of mobile-robot control to implement search motion........280

Trukhan S., Nedzved A., Ablameyko S. Region growing segmentation of CT-image by the analysis of 3D local neighborhood 285

Viattchenin D.A., Damaratski A., Nikolaenya E., Shyrai S. An outline for an approach to automatic labeling for interpretation of heuristic possibilistic clustering results 290

Viattchenin D.A., Damaratski A., Yaroma A. New parameters for relational heuristic algorithms of possibilistic clustering 295

Volkovich A. Use of color characteristics and gradient field data in disparity map building process 300

Voronov A., Doudkin A. General covering algorithm in software system for input data preparation for single-beam VLSI layout generator 305

Yankovskaya A., Yamshanov A. Bases of intelligent system creation of decision-making support on road-climatic zoning 311

Yerchak M. Oral speech: how much information is hidden behind 316

Zahariev V., Petrovsky A. Multivoice text to speech synthesis system 320

Zakharov A., Tuzhilkin A., Zhiznyakov A. Algorithm of automatic reconstruction of three-dimensional objects for CAD-systems 325

Zalesky B.A., Seredin E.N. Interactive extraction of roads and rivers in low resolution or noisy satellite images 329

Author's index 334

РАСПОЗНАВАНИЕ ОБРАЗОВ И ОБРАБОТКА ИНФОРМАЦИИ (PRIP’2014)

Доклады XII Международной конференции
28–30 мая 2014 г., Минск, Беларусь

На английском языке

Ответственный за выпуск Н.А. Рудая

Подписано в печать 05.05.2004. Формат 60×84 1/8. Бумага офсетная. Гарнитура Таймс.
Усл. печ. л. 39,5. Уч.-изд. л. 32,9. Тираж 100 экз. Заказ № 16.

Издатель и полиграфическое исполнение:
Государственное научное учреждение «Объединенный институт проблем
информатики Национальной академии наук Беларуси».
Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатных изданий
№ 1/274 от 04.04.2014 г.
Ул. Сурганова, 6, 220012, Минск