Architecture of Image Processing System for Documents

Rumen P. Mironov[1] and Roumen K. Kountchev[2]

Abstract – **Architecture for document image processing, analysis and archiving is presented. The basic functions include input/output of halftone images, pre- and post-processing, filtration, compression, enhancement, 2D transformations and interpolations.**

Keywords – **image processing, software architectures, document archiving.**

I. Introduction

In connection with the fast growing of advanced calculation techniques the systems for image processing on the base of personal computers are standing topical [1], [2]. They are built modularly and include additional frame grabbers for input and output of digital images, modules for compression, transform, filtration, analysis and visualization. In [3], [4] and [5] some image processing systems are described. All of them include basic software modules for image processing [6] and hardware accelerators for increasing of computation operations speed. They are developed on the basis of common requirements, defined in [7]:

- working on standard computer platforms with different operation systems;
- working with different input/output devices;
- working in interactive mode;
- using of internal system program language;
- using of advanced image processing algorithms;
- large possibilities for upgrades and improvements.

The image processing algorithms in these systems are included as standard libraries and each of them can be used separately by the user or supervisor. It is a function-oriented approach, which isn't suitable for imitation of human visual system working.

In this paper architecture for document image processing, analysis and archiving is presented. This architecture is task-oriented and shows the complex image processing flows, as the natural human visual system is operating. On this base in the Laboratory of Video Communications and Technologies, (Faculty of Telecommunications) an object-oriented software image processing system is developed, which is used in the research work for the different science projects.

II. Architecture of Image Processing System

On Fig.1 a block-scheme of the developed architecture for document image processing system is presented. This architecture is organized as a set of the following image processing units, based on well-known and new developed algorithms [8], [9], [10], [11]:

- *Image Loading Unit (ILoadU)*. In this module functions for loading and converting of compressed and uncompressed basic image file formats (TIFF, BMP, JPEG) and the developed in [11] image compressed file format are included. *ILoadU* also contains functions for image decryption and loading from special image documents database.
- *Image Inputting Unit (IInptU)*. In this module functions for input of document images (text, photos, tables, graphics) from scanners, photo and cameras are included.
- *Image Saving Unit (ISaveU)*. In this module functions for saving of compressed and uncompressed basic image file formats (TIFF, BMP, JPEG) and the developed in [11] image compressed file format are included. *ISaveU* also contain functions for image encryption and saving in special image documents database.
- *Image Pre-processing Unit (IPrepU)*. In this module the following functions for image pre-processing are included – functions for arithmetic and geometric operations, table operations, contrast enhancements, histogram operations, linear and nonlinear noise filtration. They are used for image quality improvement or converting in a form better suited for analysis by human or a machine.
- *Image Presentation Unit (IPresU)*. In this module different image presentation models (positioning, quad-tree, pyramidal, structural) and functions for image converting are included.
- *Image Compression Unit (ICmprU)*. In this module image compression functions, based on classical algorithms for run-length, Huffman, LZW, arithmetic, scalar and vector quantization, JPEG and a new algorithm for lossless compression [12] are included.
- *Image Geometrical Transform Unit (IGeoTU)*. In this module functions for translations, scaling, interpolations, rotations, affine and perspective transformations, spatial warping and geometrical resampling are included.
- *Image Linear Transform Unit (ILinTU)*. In this module functions for 2D linear superpositions and convolutions, correlations, discrete cosine, sine, Hadamard, complex Walsh-Hadamard, Fourier and Karhunen-Loev transforms are included.
- *Image Pseudo-Color Transformation Unit (IPCTU)*. In this module functions for pseudo-color transforms in spatial and frequency area and image halftoning by adaptive and non-adaptive error diffusion are included.

[1]Rumen P. Mironov is with the Faculty of Telecommunications, Technical University of Sofia, Kl. Ohridsky Str. 8, 1000 Sofia, Bulgaria, E-mail: rmironov@tu-sofia.bg.

[2]Roumen K. Kountchev is with the Faculty of Telecommunications, Technical University of Sofia, Kl. Ohridsky 8, 1000 Sofia, Bulgaria, E-mail: rkountch@tu-sofia.bg.

- *Image Analysis Unit (IAnlyU)*. In this module functions for image segmentation (contour, brightness, texture) morphological image processing (binary, gray), edge detection and feature extractions are included.
- *Image Post-Processing Unit (IPostU)*. In this module functions for adaptive and non-adaptive, linear and non-linear filtration of gray level or binary images are included.
- *Image Visualization Unit (IVizuU)*. In this module functions and drivers for visualization of document images are included.
- *Image Printing Unit (IPrntU)*. In this module functions and drivers for printing of documents are included.

For properly working of the developed software system the following additional modules are necessary:

- *System Supervisor*. In this module the basic functions for system control, diagnostic and interaction between separated components and operation system are included. In the system supervisor analysis and processing of system error messages and receiving of context help information are also achieved.
- *Graphical User Interface (GUI)*. In this module graphical oriented system with menus, dialogs, windows, icons, buttons, fonts and etc. is included, which can be used for dialog input and output of parameters or images, interactive processing and analysis, visualization of documents, graphical presentation of the results from the analyses.
- *Communication Module (CM)*. In this module functions for connection between the *System Supervisor* and the external modules and functions of system interpreter are included. The usage of *Communication Module* facilitate the building of processing tasks – also including new ones, which are not defined in the system. This is a way to create *External Algorithms*, established by the operator.
- *System Peripheral Module*. In this module the system peripheral drivers are situated.

Te system supports two types of data objects: image data objects and image-related, non-image data objects. A system image data object is a multi-dimensional collection of pixels, whose structure is:

- horizontal space index ($0 \le x \le 1024$);
- vertical space index ($0 \le y \le 1024$);
- depth space index ($0 \le z \le 1024$);
- temporal index ($0 \le t \le T_{max}$);
- color or spectral band index (($0 \le fz \le F_{max}$);

The system utilizes the following pixel data types: Boolean, Unsigned Integer, Signed Integer, Real and Complex. The precision and data storage format of pixel data is implementation dependent.

It supports several image related, non-image data objects. These include:

- *Chain*: an identifier of a sequence of operators;
- *Composite identifier*: an identifier of a structure of image arrays, lists and records;
- *Histogram*: a construction of the counts of pixels with some particular amplitude value;
- *Lookup table*: a structure that contains pairs of entries in which the first entry is an input value to be matched and the second is an output value;
- *Matrix*: a two-dimensional array of elements that is used in vector-space algebra operations;
- *Neighbourhood array*: a multi-dimensional moving window associated with each pixel of an image;
- *Pixel record*: a sequence of across-band pixel values;
- *Region-of-interest*: a general mechanism for pixel-by-pixel processing selection;
- *Static array*: an identifier of the same dimension as an image to which it is related;
- *Value list*: a collection of pairs of elements in which the first element is a pixel coordinate and the second element is an image measurement.

III. Experimental Image Processing Tasks

The functionality of the human visual system can serve as a reliable guide for breaking up the complex image processing tasks. First, the optical system forms an image of the observed documents. A sensor converts this image into a form that is usable for digital processing with a computer system. The first processing step, denoted as low-level image processing, enhances, restores, or reconstructs the image formed. Further processing extracts features from the images that finally lead to the identification and classification of the objects in the images and can be saved in document database. In this way, the circle is closed, leading us from documents that are converted into images and processed back to their recognition and description.

On Fig.2 the basic graphical window of the developed system with one sample test grayscale image (with size 256x256, 8 Bpp on the left side), your histogram and the same image after segmentation with threshold 138 (on the right side) are shown.

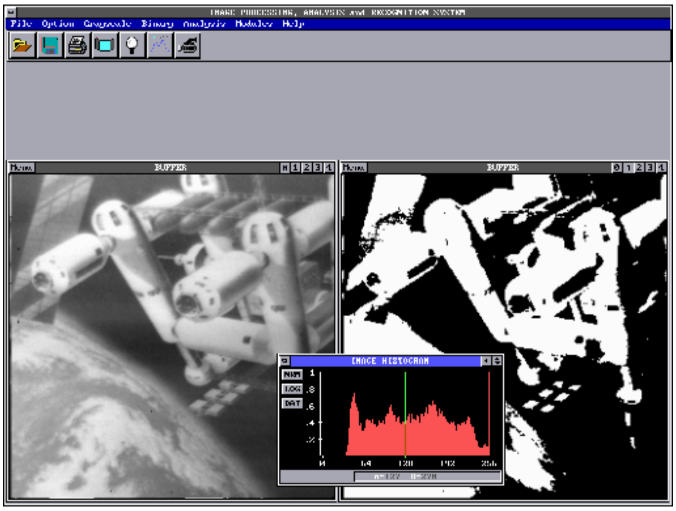

Fig. 2. Basic images ateachot37.bmp and ateachot44.bmp (320x240 pixels, 8 bits)

Using the developed system we can define various image processing tasks, which can be described semantically by the following way:

- *IInptU -> IPrepU -> IPresU -> IPostU -> ISaveU* . Inputting an image from scanner or digital photo camera, enhance image quality by pre-processing, presented in matrix form, post-processing and saving in the database. This semantic chain can be used for preliminary record of uncompressed documents in the database.
- *IInptU -> IPrepU -> IPresU -> ICmprU -> ISaveU* . Inputting an image from scanner or digital photo camera, enhance image quality by pre-processing, presentation in matrix form, compression and saving in the database. This semantic chain can be used for preliminary record of compressed documents in the database.
- *ILoadU -> IPrepU -> IPresU -> IPostU -> IVizuU* . Loading an image from file or database, enhance image quality by pre-processing, presentation in matrix form, post-processing and visualization on the screen. This semantic chain can be used for visualization of saved documents in the database.
- *ILoadU -> IPrepU -> IPresU -> IPostU -> IPrntU* . Loading an image from file or database, enhance image quality by pre-processing, presentation in matrix form, post-processing and printing on the laser or ink-jet printer. This semantic chain can be used for printing of saved documents in the database.

IV. CONCLUSION

The presented architecture, control modules, data objects and interfaces build new task oriented image processing system, which allow this system to imitate a natural human visual system. The developed architecture has the following advantages:

- module design, based on image processing tasks;
- possibilities for working in automatic and interactive mode;
- using of internal system program language;
- using of advanced image processing algorithms;
- large possibilities for upgrades and improvements.

The developed software system can be used in different areas – metallography, quantitative microscopy, analysis of biomedical images, biotechnologies, robotics, ecological monitoring, visual control in the industry, medicine, science researches and etc.

ACKNOWLEDGEMENT

The authors thank the National Fund for Scientific Research of the Bulgarian Ministry of Education and Science for the financial support by the contract VU-I-305/2007.

REFERENCES

[1] Maher A., Sid-Ahmed, *Image Processing: Theory, Algorithms, and Architectures*. McGaw-Hill, Inc., 1995.

[2] P. Stucki. *Advances in Digital Image Processing*. N. J. Plenum Press, 1979.

[3] R. Kountchev, R. Mironov. “Program System for Metallographic Image Processing and Analysis”. *XXX Science Session “Communications, Electronics and Computer Systems’95*, Sofia, May 1995, (in Bulgarian).

[4] R. Kountchev, R. Mironov. “System for Image Processing, Analysis and Recognition”. *Journal for Automatic and Informatics*, Sofia, pp.41-44, Number 4, 1996 (in Bulgarian).

[5] R. Mironov, N. Sirakov, F. Muge. “An Architecture of Virtual Multimedia Library”. *Proc. of V Ibero – American Symposium on Pattern Recognition*, SIARP’2000, Lisbon, pp.103-111, September 11-13, 2000.

[6] H.R. Myler, A.R.Weeks. *Computer Imaging Recipes in C*. Prentice Hall, Englewood Cliffs, N.J, 1993.

[7] Shi-Kuo Chang. *Principles of Pictorial Information Systems Design*. Prentice-Hall, 1989.

[8] C. Ware. *Information Visualization – Perception for Design*. 2nd Ed., Morgan Kaufmann, 2004.

[9] B. Jähne. *Practical Handbook on Image Processing for Scientific and Technical Applications*. 2nd Ed., CRC Press, 2004.

[10] W. K. Pratt. *Digital Image Processing*. 4th Ed., John Wiley & Sons, 2007.

[11] R. Kountchev, V. Todorov. “File Format organization for effective Still Image Transfer with IDP”. *37th Intern. Scientific Conf. on Information, Communication and Energy Systems and Technologies*, Proc. Vol. 1, pp. 287-290, Nis, Yugoslavia, Oct., 2002.

[12] R. Kountchev, V. Todorov, R. Kountcheva. “Lossless Image compression with IDP”. *Proc. of the Intern. Scientific Conf. on Information, Communication and Energy Systems and Technologies (ICEST’04)*, pp. 127-130, Bitola, Macedonia, 2004.

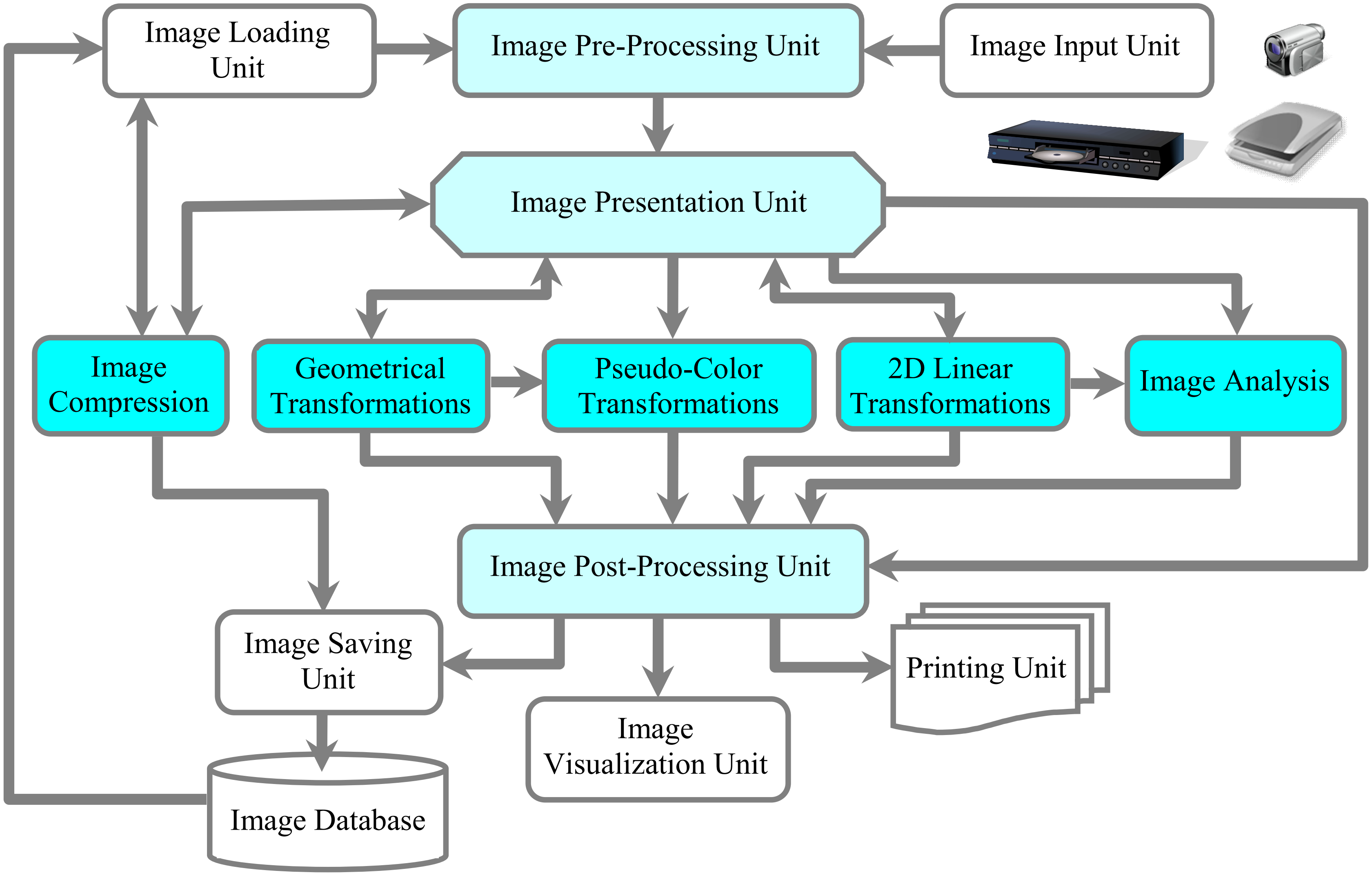

Fig.1. Architecture of Document Image Processing System