MTES: Visual Programming Environment for Teaching and Research in Image Processing

JeongHeon Lee, YoungTak Cho, Hoon Heo, and OkSam Chae

Department of Computer Engineering, KyungHee University,
Seochun-ri, Kiheung-eup, Yongin-si, Kyunggi-do, South Korea
opendori@paran.com, greizen@vision.khu.ac.kr, hhoon@naver.com,
oschae@khu.ac.kr

Abstract. In this paper, we present a visual-programming environment for teaching and research referred to as "MTES". MTES(Multimedia Education System) is the system designed to support both lecture and laboratory experiment simultaneously in one environment. It provides tools to prepare on-line teaching materials for the lecture and the experiment. It also provides a suitable teaching environment where lecturers can present the online teaching materials effectively and demonstrate new image processing concepts by showing real examples. In the same teaching environment, students can carry out experiments interactively by following the online instruction prepared by lecturer. In addition to support for the image processing education, MTES is also designed to assist research and application development with visual-programming environment. By supporting both teaching and research, MTES provides an easy bridge between learning and developing applications.

1 Introduction

Image processing plays important roles not only in the areas of object recognition and scene understanding but also in other areas including virtual reality, databases, and video processing[1][2]. With the growing demand for engineers with knowledge of image computation, the number of schools with undergraduate elective courses on topics related to image computation, such as image processing and computer vision, is increasing rapidly and many educators are trying to find an effective method of teaching such courses. Some faculties even tried to integrate the image computation problems into the core curriculum as examples. However, the image computation involves the image, which is difficult to describe verbally, and the theory is defined on the strong mathematical basis. In the previous studies on image processing education, researchers point out the followings [3]:

- The image processing theory can be explained most effectively by visual means.
- It is necessary to complement the theory with computer-based experiments. Some researchers found that the laboratory exposure before the theoretical

V.S. Sunderam et al. (Eds.): ICCS 2005, LNCS 3514, pp. 1035 – 1042, 2005.

treatment begins is most effective. It helps students understand the theory more clearly and easily.
- Hands-on experiment is essential to demonstrate the image computation concepts through examples.
- Image processing industries want to hire the students exposed to extensive laboratory work to be prepared for real applications.

The researchers also recommend to develop courses more accessible to students with computer science background who are more familiar with computer programming than applied mathematics, which is the basis of most image processing and pattern recognition theory.

Taking these into consideration, many researches have been done to develop the teaching environment that many lecturer and students in the field of image computation want to have. Those researches can be divided into two groups. The first group is the library systems with simple execution tools such as menus and script languages. The library consists of visualization functions, basic image processing functions, and interface functions defined to access image data [2][4][5]. These library systems are designed to simplify laboratory experiment by providing basic functions frequently required for the experiment. However, these library systems do not provide the programming environment with which student can implement and analyze their ideas without writing complex programs. The researches in the second group tried to solve these problems by taking advantages of the general research tools [2][6-10]. These systems are designed to support research and education. They are equipped with simple programming tools, such as a visual-programming tool and an interpreter, and various image analysis tools. With those, users can easily write a code to test their idea. However, most of these systems are designed for idea verification not for algorithm development. The structures of image processing functions in those systems are not transparent to the users [2]. And none of these systems provide either the tools to prepare online materials for lecture and experiment or the tools to present them simultaneously in the same environment.

In this paper, we propose a new visual-programming environment, referred to as "MTES", supporting both lectures and laboratory for image computation. MTES provides lecturers tools to create and register viewgraphs for the lectures and the environment to present them to the students with real examples. It is also designed as an efficient environment for research and application development by supporting the development of true object-oriented algorithm in image processing, systematic algorithm management, and easy generation of application.

To support easy creation of reusable image processing algorithms, MTES provides data classes with resources for essential image processing operations. With class libraries and online function management tools, users create an image processing software component that can be recognized as an icon on a function icon window. To promote the reuse of the user-defined functions, MTES provides an intelligent visual-programming environment which allows users to create their own image processing applications by simply dragging functions from a function icon window and dropping

to a visual workspace. For the transparency of codes, the source code of the data class library and basic image processing functions pre-registered in MTES are open to users.

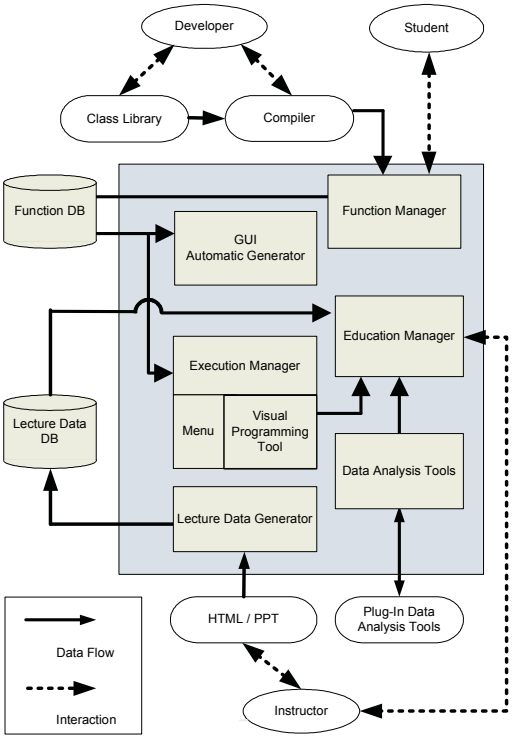

Fig. 1. Overall structure of MTES

2 Design of Visual Programming Environment for Teaching and Research of Image Processing

The requirements for an ideal visual-programming environment for teaching and research have been studied by many researchers and some of them are summarized in the previous section. In this paper, we present a new visual-programming environment designed to best fit for those requirements. In the process of designing the system, we tried to meet the following goals.

- Lecture should be given with real world examples supported by various analysis tools that can visualize the concept of the theory in the lecture.
- Lecture and hands-on experiment for the lecture should be carried out simultaneously in the same environment.
- Students should be able to easily program and test their algorithms.

- Students should be able to create new applications by making use of user-defined functions and existing functions.
- The environment should provide instructors an easy way to prepare and register lecture materials.
- The environment should be easy to learn so as to minimize “learning overhead”.
- The environment should support both research and teaching.

The overall structure of the system, designed to meet the above goals is shown in Fig. 1. The proposed system can be divided into three parts: education module, algorithm generation and management module, and algorithm execution module. The education module provides instructors tools to generate the teaching materials for both lecture and laboratory experiments. The lectures and experiments are tightly integrated. The algorithm generation module enables user to create reusable user-defined functions by making use of commercial compiler and the image class library, which is defined to generate sharable image processing algorithm. The user-defined functions can be registered and maintained with all the information necessary to make reuse of them. The algorithm execution module executes user-defined functions and pre-registered basic image processing functions. In MTES, all the functions are represented as icons organized as a hierarchical tree. The functions are executed individually by menu or in a group by using the visual-programming environment.

3 Education Module

To provide students the best image processing education, the teaching material consisting of the lecture viewgraphs, examples, and hands-on experiments should be integrated to an e-book and presented to students interactively. The educational environment is divided into two parts: a lecture generation module and a learning module. The lecture generation module provides instructors the tools necessary to create the interactive teaching material. The learning module enables students to study new theory with the on-line lecture viewgraphs, interactively follow through the examples, and implement a new algorithm and apply it to solve real world problem.

To create the integrated teaching material, a lecturer prepares lecture viewgraphs and examples separately by using the commercial software of viewgraph generation like MS-PowerPoint and MTES’s visual-programming tool. And then integrate them in the form of an e-book as shown in Fig. 2. MTES provides the functions necessary for the integration as shown in the pop-up menu in Fig. 2.

The teaching material may have the structure of the conventional paper-based textbook as shown in Fig. 2. However, a chapter is defined to cover one week of class work. Each chapter consists of lecture, example, and practice all of which can be covered in a week. Fig. 3 shows a lecture screen with the example menu activated. Student and instructor go through the viewgraphs to study a new topic and then students can carry out experiments interactively by either selecting an example on the example menu or a practice problem on the practice menu.

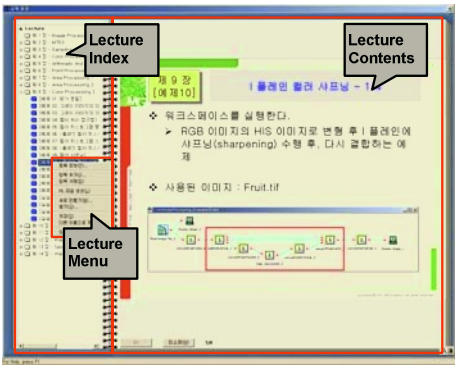

Fig. 2. Lecture screen with the command menu

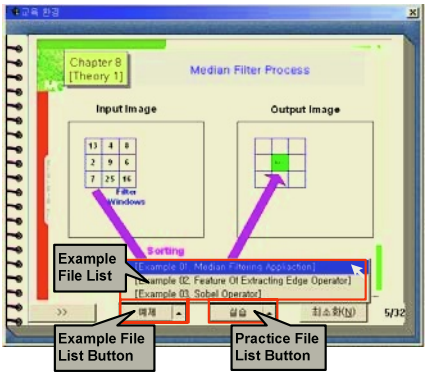

Fig. 3. Lecture Screen

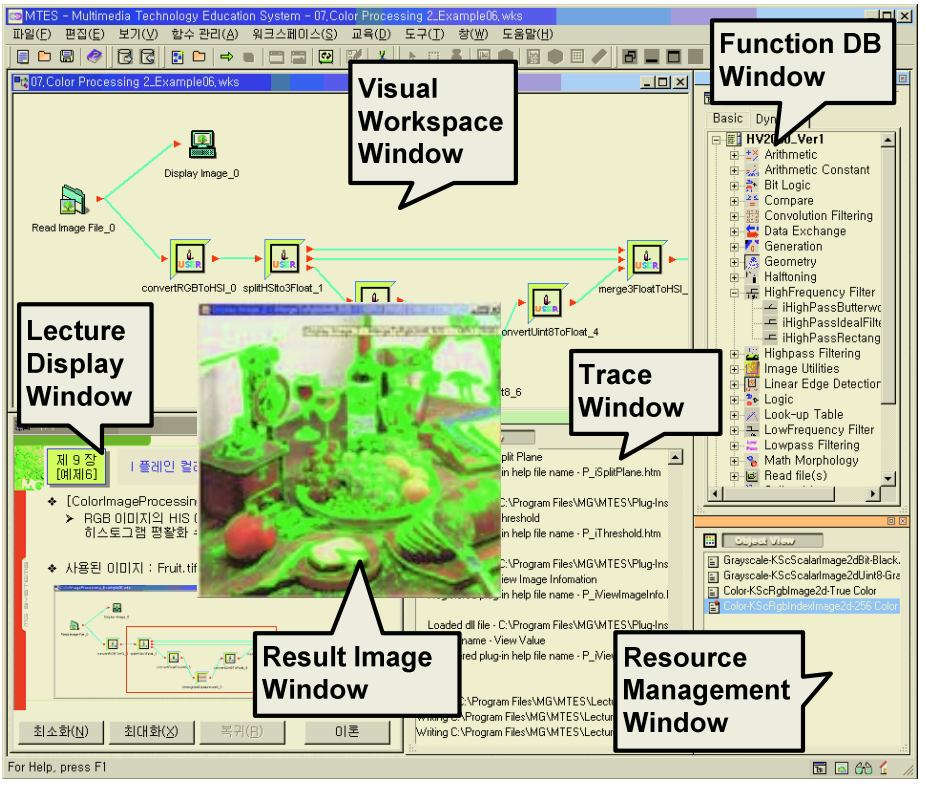

Fig. 4. Laboratory Experiment Screen

In the experiment mode, MTES displays one or more of the following three windows: instruction window, visual-programming window, and MS Visual Studio.

The instruction window displays viewgraphs showing the detail procedure of the experiment to guide the experiment as shown in Fig. 4. The visual-programming window loads the visual program that will be needed in the experiment. For the experiment involving the programming of new algorithms, MTES activates the Visual Studio loaded with the source code that includes all the codes except the main body of the algorithm. Student can complete the program by typing the main logic of the algorithm, compile, and execute it with the visual program loaded in the visual-programming window. By taking the burden of writing the house-keeping code, which dealing with the image I/O and memory management, out of students, they can concentrate all their efforts on developing the logic of image processing algorithm. Users can take advantage of MTES's algorithm-development environment to create and test user-defined functions. They can write their own image processing algorithms by using the commercial compilers such as MS visual C++. And generate a new application program by combining them with the functions registered in the function database in the visual-programming environment. The result of the application program is analyzed by using various image analysis tools in the system. For instructors, this mode could be used to demonstrate new image processing concepts with real examples before the theoretical lecture begins.

In MTES, the experiment is divided into "**example**" and "**practice**". The **example** guides a student step by step until he reaches the result. However, the **practice** presents the problem description and essential information to solve the problem and let the student do the rest.

4 Algorithm Generation and Execution Module

To support both education and research, MTES allows users to create their own functions by using commercial programming compiler such as Microsoft Visual C++ and accumulate them systematically for reuse. To simplify the generation and management of sharable user-defined functions, it provides the library of image-data class, which provides resources for essential image processing operations, and the algorithm manager that handles the registration and management of user-defined functions. The algorithm manager maintains not only the function library but also the information necessary for the reuse of the functions.

Fig. 5 shows the user interface of the algorithm manager. Through this interface, we can define a new function by inputting the name and parameters of the function to be defined. We can also register the help file required for the online help generation, the visual program needed to test the function, and the source code of the function. The visual program is displayed as a button on the online help window. User can test the function by simply clicking the button.

When the registration procedure is completed, the system registers a new function icon on the function database and creates a project file consisting of a source file and header file to generate the DLL file for the function. The function will be ready as soon as we fill in the main body of the source code and compile. If we want to modify the function later, we can reload the source code by selecting the function on the

function icon window and execute a modify command on the pop-up menu. It will reload the project file for the function. All that we need to do is "modify and compile."

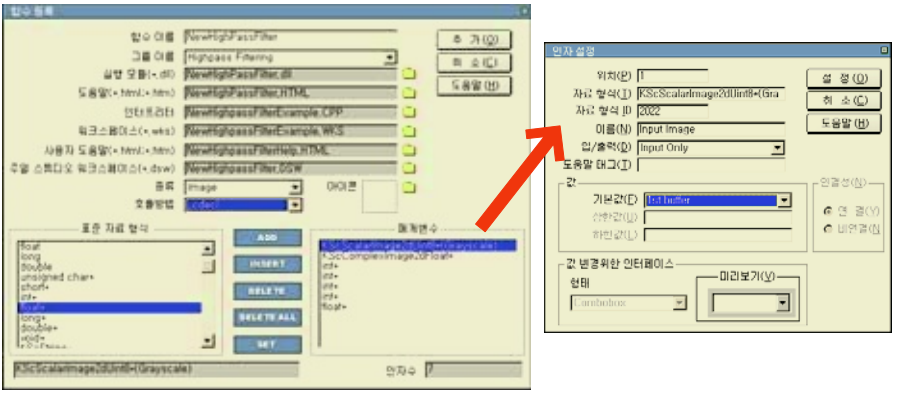

Fig. 5. Function Registration

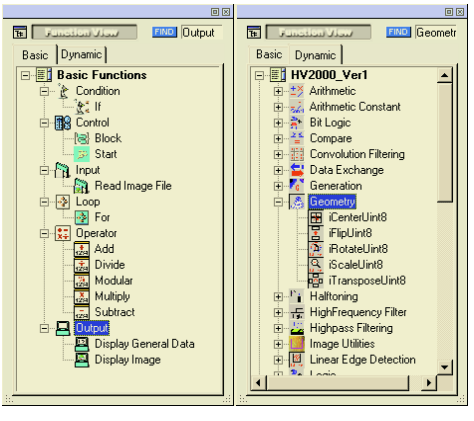

Fig. 6. Function DB Window

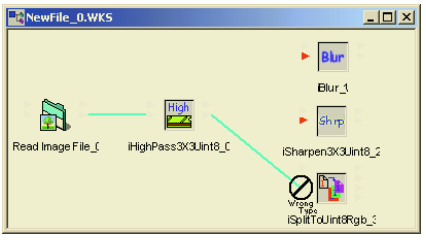

Fig. 7. Compatibility Checks

The command(function) DB window of visual program displays the list of basic commands and I/O commands for visual programming as shown in Fig. 6. The visual-programming window is the workspace that allows users to create image processing applications by connecting visual command icons and user-defined function icons in the function window. Unlike most of existing visual-programming environment, which troubleshoots parameter compatibility in run time, the proposed system troubleshoots the parameter compatibility while connecting icons for programming. This allows less chance of error in run time. Fig. 7 shows the highlighted input nodes compatible for the output node to be connected.

5 Conclusions

In this paper, we present a visual-programming environment for image processing that supports both research and education. It helps lecturers to generate online teaching materials consisting of viewgraphs for a lecture and the direction of the experiments needed for the lecture. The system also provides a suitable teaching environment to present the lecture and experiment simultaneously in the same environment.

This system has been used to teach graduate and undergraduate image processing courses in over 13 universities in Korea. From our experience in using this system in teaching image processing courses for last 4 years, we found that it helps students to understand complex image processing concepts by showing real examples during the lecture. It also helps them to be well prepared for real world image processing applications by integrating the hands-on experiment to the lecture.

References

1. G. J. Awcock and R. Thomas, “The Teaching of Applied Image Processing at thee University of Brighton,” The institution of electrical engineers. Savoy Place, London WC2R 0BL, UK, 1993
2. Gregory W. Donhoe, and Patrick F, Valdez, “Teaching Digital Image Processing with Khoros,” IEEE Trans. on Education, Vol. 39, No, 2, pp.137-142, 1996
3. K. Bowyer, G. Stockman, and L. Stark, “Themes for Improved Teaching of Image Computation,” IEEE Trans. on Education, Vol. 43, No. 2, 2000
4. J. A. Robinson, “A Software System for Laboratory Experiments in Image Processing,” IEEE Trans. on Education, Vol. 40, No, 4, pp.455-459, 2000
5. J. Campbell, F. Murtagh, and M. Kokuer, "DataLab-J: A Signal and Image Processing Laboratory for Teaching and Research," IEEE Trans. on Education, Vol. 44, No. 4, pp.329-335, 2001
6. C. S. Zuria, H. M. Ramirez, D. Baez-Lopez, and G. E. Flores-Verdad, “MATLAB based Image Processing Lab Experiments,” FIE Conf., pp.1255-1258, 1998
7. R. Lotufo, R. Jordan, “WWW Khorosware on Digital Image Processing,” Brazilian Conf. on Computer Graphics and Image Processing, Sibgraphi, 1995
8. H. Bassman and P. Besslich, “Ad Oculos: Digital Image Processing/Book and Software”
9. S. E. Umbaugh, So.Illinois Univ., Edwardsville, Illinois, “Computer Vision and Image Processing: A Practical Approach Using CVIPTools,” Prentice Hall.
10. W. Rasband, NIH Image, available online at URL http://rsb.info.nih.gov./nih-image