Implementation of Image Filter for Computer Graphic Software

Jang Seung-Ju, Kim Hyung-Bo**, Yoon Sang-Bok***

*Dong-eui University, Department of Computer Engineering, Busan, Korea
**Dong-eui University, Department of Urban Engineering, Busan, Korea

Summary

An object of developing computer graphic S/W in this paper is, first of all, to solve manpower problem which is faced by printing and advertisement companies. The system developed in this paper enables reducing manpower by minimizing existing package design processes which are based on labor intensively and by having all works performed by computers. The system also promotes working effects of printing and advertisement designs as well as eliminates possible errors between what is requested by the user and what is actually produced before final products are made by presenting accurate sample products for advertisers. Compared to conventional method which requires a great deal of time and manpower, the present system is effective economically and timewise. Further, while conventional printing and advertisement systems are very expensive, the present system may be developed and supplied at a low cost. Using the present system for most printing and advertisement companies implies reducing expenditure for printing and advertisement.

Use the Image Filter Set to adjust image sharpness and edges, as well as its brightness, contrast and color. Each image filter parameter works as a mathematical function through which the result image is affected. There are various kinds of image filters. In this paper deals with smoothing filter, edge detector, which is point, line, prewitt, sobel, laplacian detector. To implement of this filter into image editor, I use Visual C++ programming language. These image filters is very usefully utility in image editor software.

In the fields of printing and advertisement, most processes prior to printing are performed manually for small businesses, and expensive Macintosh are used for large-scale businesses for the purpose of printing. PC have been used widely recently, but there have been limits that graphic S/W that can be used in PC are insufficient or using them is too expensive for small and medium businesses. Therefore, it is critical to develop graphic S/W which are not costly but equipped with necessary functions for small and medium businesses. Development of such graphic S/W is effective for reducing printing cost.

Key words:
Image Filters, Smoothing Filter, Edge Detector, Image Editor

Introduction

An object of developing computer graphic S/W in this paper is, first of all, to solve manpower problem which is faced by printing and advertisement companies. The system developed in this paper enables reducing manpower by minimizing existing package design processes which are based on labor intensively and by having all works performed by computers. The system also promotes working effects of printing and advertisement designs as well as eliminates possible errors between what is requested by the user and what is actually produced before final products are made by presenting accurate sample products for advertisers. Compared to conventional method which requires a great deal of time and manpower, the present system is effective economically and timewise.

The content and scope of this paper for the development of computer graphic S/W for perspective package design are as follows:

- ◆ Definition and design of user interface necessary for plan construction;
- ◆ S/W functions providing with high-resolution function;
- ◆ Editing and debugging function of plan prior to output;
- ◆ User interface expression function enabling feeling of reality;
- ◆ Linking function with printing using the plan and product manufacture system; and
- ◆ The image editor was developed in this paper is called "E yes image editor"

2. Requirements for Image Editor

What is developed in this paper is a computer program used for printing and advertisement companies. Printing and advertisement processes are complicated. Among the processes, manuscript construction and proofreading process are important. At present, these processes are performed manually in small businesses. If these manual processes are converted into computerized processes, it is possible to save manpower and time. This effect has a benefit of lowering the price of products.

Generally, printing processes are comprised of the followings:

(1) Planning conference
a) Planning: confirmation of entire object,
Selection of products, distribution sites, estimate, schedule.
b) Determination of editing content and theme development:
rough sketch, thumb nail sketch, editing arrangement table, list of material collection sites, etc.

Manuscript received March 5, 2009
Manuscript revised March 20, 2009

c) Determination of printing method: confirmation of editing

Arrangement table, printing material size, number of pages, number of colors, number of editions, bookbinding paper.

(2) Selection and collection of materials

a) Collection and writing of sentence copying, request for manuscript to authors and copy writers.
b) Collection of photographing slides, hiring of cameramen, arrangement of photographing, arrangement of film library.
c) Request for production to illustrators.
d) Collection of tables and drawing plates, design.

(3) Editing background layout

a) Organization and selection of collected manuscripts, comparison with original materials, make-up, photograph plates, illustration, checking of development of rough sketches, conference with related personnel.
b) Designation of editing arrangement prior to sending manuscripts to printing shop, designation of manuscript, photographs, drawing plates, and illustration.

(4) Printing

a) Receipt of manuscript (printing shop): manuscript, photographs, drawing plates are received together with an editing arrangement paper.
b) Comparison and checking of designation of proofreading, manuscript, and editing assignment.
c) Revision of colors: appearance of plates, checking of size and position (confirmation of blue, red, yellow ink plates).
d) Admission to printing, admission to printing site, adjustment of color revision state.
e) Confirmation of plating materials and plating format as plating and provisional plating.

(5) Checking of delivery

a) Checking and delivery of products, checking of appearance of printing materials through comparison with revised colors, confirmation of size and volume.
b) Distribution of products to bookstores, business agencies, branch stores, reserved readers (regular readers in case of magazines), etc.

Figure 1. Advertisement/Printing Processes

Requirements for an advertisement and printing editor used during printing procedure in order to save manpower and time are defined to be as follows:

◆ Definition and design of user interface necessary for plan construction;
◆ Providing high-resolution function;
◆ Editing and debugging of plan prior to output;
◆ User interface enabling feeling of reality;
◆ Linking function with printing using the plan and products to be manufactured.

2.1 Definition and Design of User Interface Necessary for Plan Construction

A user interface necessary for plan construction is a critical element. A program is not a good program if it is not convenient in learning and using for users even though it has superior functions. A user interface is defined and designed from this point of view.

(1) File Open / Save / Save As

(2) Indication of file size information on a status bar

(3) Supply for a histogram analyzer for BMP file being edited

(4) Providing with a drawing function

Draw Lines / Draw Rectangles / Draw Rounded Rectangles /Draw Ellipses / Draw Arcs / Draw with Pencil / Draw with Brush / Fill a Region

(5) Editing of an image file

Select Rectangle Region

- Cut / Copy / Paste

Select Polygon Region

(6) Insert Text into Image

(7) Image Amplification and Reduction (Zoom In/Out)

(8) Image Eraser Function

(9) Image Modulation Function (filter)

Smoothing Filter with a Median Method / Laplacian Operator / Line Detection / Point Detection / Prewitt Operation / Sobel Operator / Dilation / Erosion / Bright Correct / Contrast Correct / Edge Detection / Emboss / Ep Smooth / Flip X / Flip Y / Halftone / High Pass / Warping

3. Design Image Filter

<Image Modulation Function (filter) >

◆ Smoothing Filter

Smoothing is mainly used for eliminating noise included in input image. There are many methods of smoothing black-and-white images. The simplest method is replacing a value obtained by adding nine pixel values and dividing it by 9 with a new value. If a value larger than 9 is used, the smoothing effect becomes greater. Another method is Gaussian smoothing method in which Gaussian function is used in smoothing. However, these methods have a problem of eliminating edge regions when noise of input images is eliminated. In order to solve this problem, there is a method of smoothing while edge regions are retained. The method of boundary preservation smoothing is shown in Equation (1) below:

$$I^{(t+1)} = \frac{1}{N^{(t)}(X)}(I^{(t)}(x,y-1)W^{y^{(t)}}(x,y-1)+I^{(t)}(x-1,y)W^{x^{(t)}}(x-1,y)$$

$$+I^{(t)}(x,y)+I^{(t)}(x+1,y)W^{x^{(t)}}(x+1,y)+I^{(t)}(x,y+1)W^{x^{(t)}}(x,y+1)$$

$$N^{(t)}(X) = W^{y^{(t)}}(x,y-1)+W^{x^{(t)}}(x-1,y)+1+W^{x^{(t)}}(x,y)+W^{y^{(t)}}(x,y)$$

$$x^{(t)}(x,y) = \exp(-\frac{\left|I^{(t)}(x+1,y)-I^{(t)}(x,y)^2\right|}{2k^2})$$

$$y^{(t)}(x,y) = \exp(-\frac{\left|I^{(t)}(x,y+1)-I^{(t)}(x,y)^2\right|}{2k^2})$$

(1)

When Effect - Smoothing Filter item is selected from the menu as one of the tools providing with the filtering special processing effect in this program which is actually implemented, sub-menus will appear. Presently, only Median function is provided. Since filtering is accomplished by using a median value, i.e., an average, entire image is processed to have soft texture. Such function is widely used for rough photographs or in expressing images having soft texture for other special purposes.

◆ Edge Detector

Edge detection is a method of locating the boundary of an object in an image or the boundary line enabling distinguishment of adjacent objects by identifying pixels of which color values or gray-levels are significantly different from those of neighboring pixels. Edge detection is divided into two steps of obtaining edge strength showing the degree of edging at the position of each pixel of input images and of determining to what edge strength is to be considered.

At present, the widely used methods of finding edge strength in black-and-white images include the gradient method having Roberts cross-gradient operators, Prewitt operators, Sobel operators. and Laplacian method. Roberts cross-gradient operators, Prewitt operators, and Sobel operators use a gradient which is defined in Equation (2) as follows:

$$\nabla F = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

(2)

The edge strength is defined in terms of the size of gradient as shown in Equation (3) below:

$$\nabla f = mag(\nabla F) = \left[(\frac{\partial f}{\partial x})^2 + (\frac{\partial f}{\partial y})^2\right]^{1/2}$$

(3)

(i) Roberts cross-gradient operator

The edge strength can be obtained from Equation (4) as follows:

$$\nabla f = \left[(Z_5 - Z_9)^2 + (Z_6 - Z_8)^2\right]^{1/2}$$

$$= \left|Z_5 - Z_9\right| + \left|Z_6 - Z_8\right|$$

(4)

The above equation is implemented by the length 2 x 2 mask as in Figure 2(b).

(ii) Prewitt operator

The edge strength can be obtained from Equation (5) as follows:

$$\nabla f = \left|(Z_7+Z_8+Z_9)-(Z_1+Z_2+Z_3)\right| + \left|(Z_3+Z_6+Z_9)-(Z_1+Z_4+Z_7)\right|$$

(5)

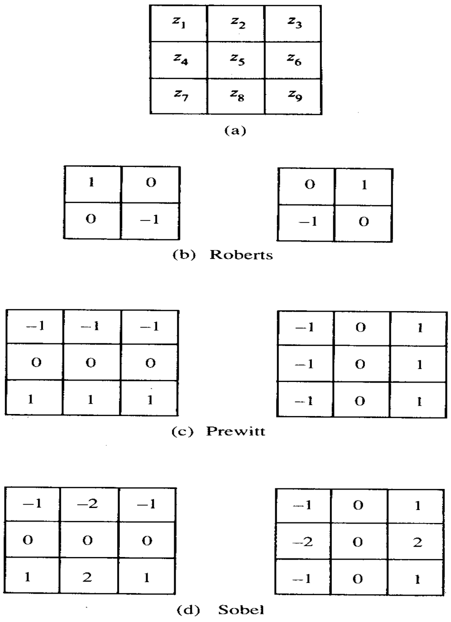
The absolute value of the front portion in Equation (5) shows the difference in x direction, while the absolute value of the rear portion shows the difference in y direction. It is implemented as in Figure 2(c).

(iii) Sobel operator

This operator shows the most excellent performance among edge operators, and the edge strength can be obtained as in Equation (6) below:

$$\nabla f = |(Z_7 + 2Z_8 + Z_9) - (Z_1 + 2Z_2 + Z_3)| + |(Z_3 + 2Z_6 + Z_9) - (Z_1 + 2Z_4 + Z_7)| \tag{6}$$

It is implemented as in Figure 2(d).



z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

(a)

1	0
0	−1

0	1
−1	0

(b) Roberts

−1	−1	−1
0	0	0
1	1	1

−1	0	1
−1	0	1
−1	0	1

(c) Prewitt

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

(d) Sobel

Figure 2 Edge operators

(iv) Laplacian operator

Laplacian is a function defined by the secondary differentiation as shown in Equation (7) shown below:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{7}$$

It is possible to implement this function as shown in Equation (8) below by using filter:

$$\nabla^2 f = 4Z_5 - (Z_{52} + Z_4 + Z_6 + Z_8) \tag{8}$$

However, Laplacian operator is not widely used because of several difficulties.

The edge operator implemented in the present program is a little modified from an existing method so that it can be used for color images. The edge strength can be obtained according to the following equation since each pixel in color images has R, G, B values:

```
gr =    (float)1/(float)4*(DIST_3D(pixel[1],pixel[7]) +
       (float)2*DIST_3D(pixel[2],pixel[6]) +
        DIST_3D(pixel[3],pixel[5]));
gc =    (float)1/(float)4*(DIST_3D(pixel[7],pixel[5]) +
        (float)2*DIST_3D(pixel[4],pixel[0]) +
        DIST_3D(pixel[3],pixel[1]));
g =     (float)sqrt(gr*gr + gc*gc);
```

where gr is the widthwise difference, gc is the lengthwise difference, and g is the edge strength. Pixel [1] - pixel [8] refer to nine pixels. Eventually, the edge strength is obtained by using Sobel operator. The process of finding a pixel corresponding to an edge which is the second step in edge detection is obtained by thresholding with 70 as a critical value in the present program.

This function can be found in sub-menus in the developed program when “Effect - Edge Detector” item is selected. At present, the functions for all sub-menus are implemented. They show the effect of detecting and filtering related pixels as in one boundary. If these functions are properly used, ordinary photographs can be modulated as if they were sketches drawn by artists.

4. Design Interface and Implementation of EYES Image Editor

Design and implementation of detailed functions with respect to requirements for an image editor described above are as follows:

4.1 File Open / Save / Save As

This is the most basic function for all Window programs. Generally, the basic function is provided by Visual C++ language. However, if the basic function is not sufficient, functions should be extended. Eyes program has additional functions inserted to the functions provided basically. In case of File Open, it is possible to process BMP file. In File Save / Save As, BMP file format is stored.

4.2 Indication of file size information on a status bar

The size of bit map being used (width x length), the number of bits per pixel, i.e., the number of bits determining the number of colors, are indicated.

4.3 Supply for a histogram analyzer for BMP file being edited

A histogram is a tool used by all graphic users and is a function provided by most graphic tools. It is indicated as a value up to 0.255 in the gray-level of each RGB in Eyes program, and shows luminance through analysis of pixel unit for the image.

4.4 Drawing function

The drawing function in Eyes program enables drawing of lines, rectangles, rounded rectangles, ellipses, arcs, etc. Tools for drawing include pens, brushes, etc. Most editing functions provided by Eyes are made by using a class provided by MFC (Microsoft Foundation Class) library. Particularly, LineTo( ), Rectangle( ), and Arc( ) provided as member functions of CDC class are used. Also, pen and brush drawings are implemented by using Cpen and Cbrush classes. To be noted is that classes such as Cline, Rectangle, etc. which are derived from an abstract class called CDrawObject are made and used in Eyes program for the flexibility of these functions. All functions in previous classes are inherited to CDrawObject class.

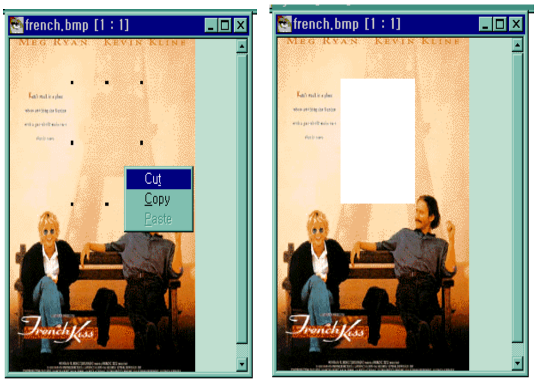
4.5 Editing of an image file

An image file editing function is a function used when an image and a region are selected and changed as desired. Selection of an image region is divided into two: one is selection of a region in a rectangular form; another is selection of a region in a polygonal form.

<Select Rectangle Region>

This function can be used by selecting “Tool - Select” in the menu or by clicking the arrow-shaped icon in the tool bar. This is a tool selected as a default when a program is initiated or a new image is opened. When a specific region is to be selected in an image, it can be accomplished by clicking the left side of the mouse at an initial stage, dragging to the desired portion, and releasing the left mouse button. At this time, the region can be moved or deleted, or its size may be controlled. The followings are the functions accomplished by this tool.

- **Cut**: This is the same function as Cut provided by any general graphic editor, in which the image of selected region is copied into the clipboard and the region is filled with a background color (white in case of default).



(a) Region selection from original image (b) Cut shape

Figure 3 Image Edition by Using Select Rectangle Region

In Figure 3(a), a clipboard menu appears when the right mouse button is pushed. The region selected by choosing Cut in Figure 3(a) is copied into the clipboard memory. The Cut region is filled with the background color. The background region is designated to be white. The result of Cut operation for the finally selected region is shown in Figure 3(b).

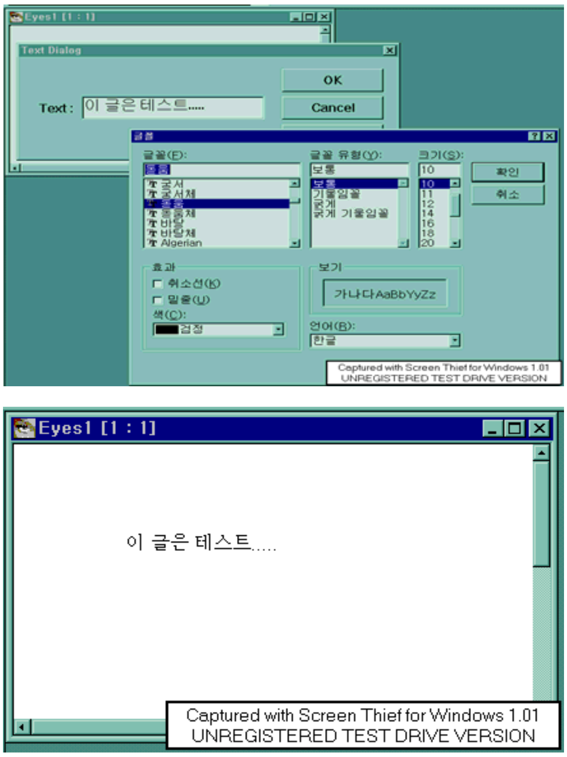
- **Copy**: It is almost similar to Cut, however, the selected region is not erased once the image is copied into the clipboard.
- **Paste**: The image data copied in the clipboard are pasted to another image region desired. However, it is not possible to change the size of image in the clipboard. These functions may be approached by using Edit in the menu or a tool bar. Also, quick menu method may be supported by using the right mouse button of the image region with respect to these functions.

<Select Polygon Region >

It is almost similar to the rectangular region. The only difference is that the region may be selected in the form of a polygon. This function can be conveniently used when a particular region is manipulated in an image.

4.6 Insert Text into Image

This function may be used by clicking an icon indicated as 'A' in the tool bar. Both Korean and English are supported, a text setting box can be obtained by clicking a region into which the text is to be inserted in the image, and the letters appear on the image as a designated option when the corresponding option is designated. The foreground and background colors are both supported.



Text : 이 글은 테스트.....

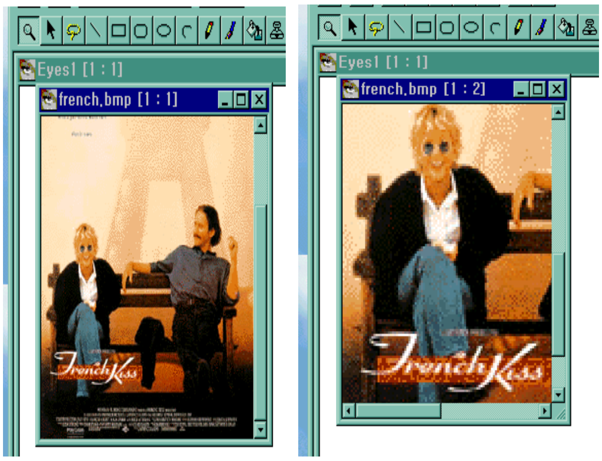
이 글은 테스트.....

Captured with Screen Thief for Windows 1.01
UNREGISTERED TEST DRIVE VERSION

Figure 4 Text Data Input Screen on Image

4.7 Image Amplification and Reduction (Zoom In/Out)

The image Zoom In/Out function may be used by clicking a spectacles-shaped icon at the far left side of the tool bar. Practically it does not magnify or reduce an image, but is used when detailed work is required for a small image or a large image needs to be seen on one screen, like a magnifier. When converted into this mode, the shape of a cursor will be changed into a magnifier shape in the corresponding image. Zoom Out is shown when the left mouse button is pushed, while Zoom In is shown when the right mouse button is pushed. For reference, the ratio between the original image and the actually viewed image will appear on the window caption bar.



(a) Original image (b) Doubly amplified image

Figure 5 Image manipulation using Zoom In / Zoom Out function

4.8 Image Eraser

An eraser function may be used in an image by clicking an eraser-shaped icon in the tool bar. The eraser function enables erasing of a portion completely as desired from the image. The desired portion may be erased by designating the size of an eraser by the mouse at the beginning as drawing a square and dragging the portion. At this time, the region to be erased is filled with an assigned color.

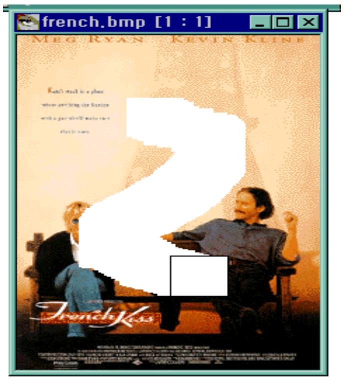


Figure 6 Image Manipulation Using Eraser

4.9 Image Modulation Function (filter)

◆ Smoothing Filter

Smoothing is mainly used for eliminating noise included in input image. There are many methods of smoothing black-and-white images. The simplest method is replacing a value obtained by adding nine pixel values and dividing it by 9 with a new value. If a value larger than 9 is used, the smoothing effect becomes greater. Another method is Gaussian smoothing method in which

Gaussian function is used in smoothing. However, these methods have a problem of eliminating edge regions when noise of input images is eliminated. In order to solve this problem, there is a method of smoothing while edge regions are retained.

When Effect - Smoothing Filter item is selected from the menu as one of the tools providing with the filtering special processing effect in Eyes program which is actually implemented, sub-menus will appear. Presently, only Median function is provided. Since filtering is accomplished by using a median value, i.e., an average, entire image is processed to have soft texture. Such function is widely used for rough photographs or in expressing images having soft texture for other special purposes.

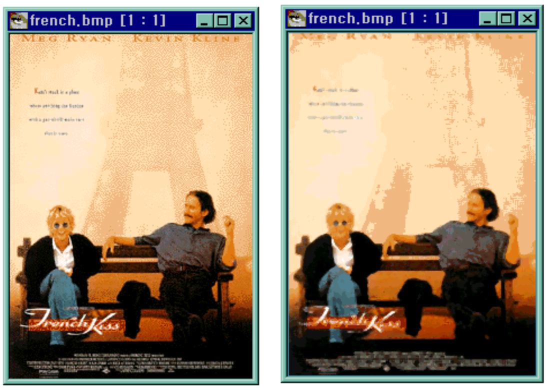
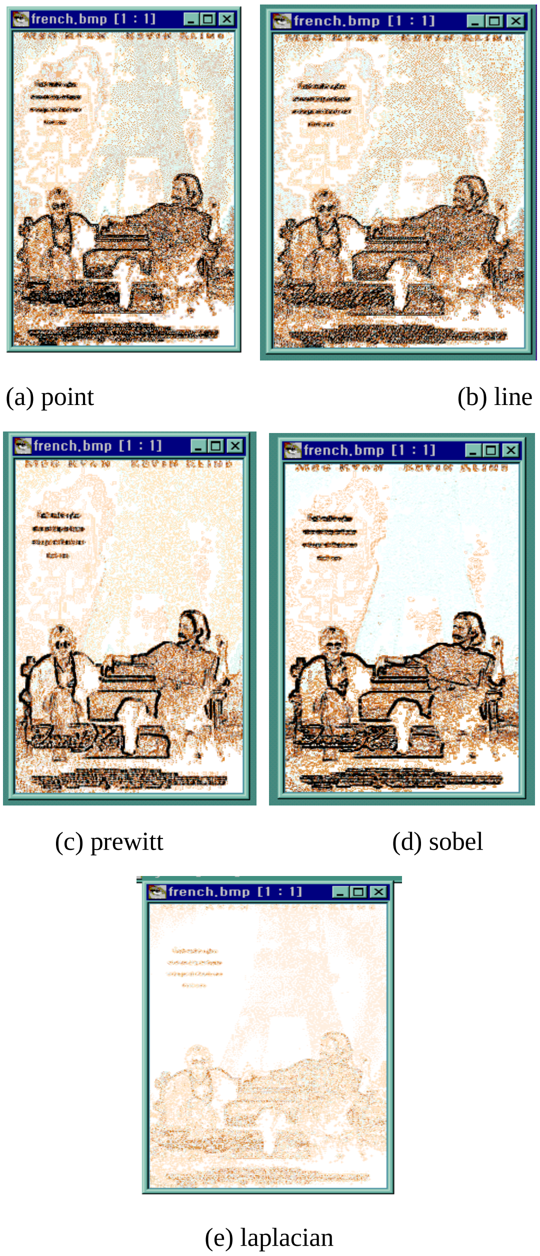


Figure 7 Image modulation using smoothing filter

◆ Edge Detector

Edge detection is a method of locating the boundary of an object in an image or the boundary line enabling distinguishment of adjacent objects by identifying pixels of which color values or gray-levels are significantly different from those of neighboring pixels. Edge detection is divided into two steps of obtaining edge strength showing the degree of edging at the position of each pixel of input images and of determining to what edge strength is to be considered. At present, the widely used methods of finding edge strength in black-and-white images include the gradient method having Roberts cross-gradient operators, Prewitt operators, Sobel operators. and Laplacian method. Roberts cross-gradient operators, Prewitt operators, and Sobel operators use a gradient.

This function can be found in sub-menus in Eyes program when "Effect - Edge Detector" item is selected. At present, the functions for all sub-menus are implemented. They show the effect of detecting and filtering related pixels as in one boundary. If these functions are properly used, ordinary photographs can be modulated as if they were sketches drawn by artists.



(a) point (b) line

(c) prewitt (d) sobel

(e) laplacian

Figure 8 Image modulation using other filters

5. Conclusion

The image filter system developed in this paper enables reducing manpower by minimizing existing package design processes which are based on labor intensively and by having all works performed by computers. The system also promotes working effects of printing and advertisement designs. Use the Image Filter Set to adjust image sharpness and edges, as well as its brightness, contrast and color. Each image filter parameter works as a mathematical function through which the result image is affected. In this paper deals with smoothing filter, edge detector, which is point, line, prewitt, sobel, laplacian detector. To implement of this filter into image editor, I use Visual C++ programming language. These image filters is very usefully utility in image editor software.

References

[1] Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Addison Wesley, 1992.

[2] Masid-Ahmed, Image Processing, McGraw-Hill, Inc., 1995.

[3] Byung-il Ahn, Print Engineering, Sejin Publishing Company, 1993.

[4] Hyung-keun Kim, Sung-hee Park, Development of graphic tools using MFC 4.0 of Visual C++4.0, Crown Publishing Company, 1996.

[5] David J. Kruglinski, Inside Visual C++ 4, Microsoft Press, 1996.

[6] Byung-sun Chun, Visual C++4.0, Samyang Publishing Company, 1996.

Seung-Ju, Jang received a B.Sc. degree in Computer Science and Statistics, and M.Sc. degree, and his Ph.D. in Computer Engineering, all from Busan National University, in 1985, 1991, and 1996, respectively. He is a member of IEEE and ACM. He has been Professor in the Department of Computer Engineering at Dongeui University since 1996. He was a member of ETRI(Electronic and Telecommunication Research Institute) in Daejon, Korea, from 1987 to 1996, and developed the National Administration Multiprocessor Minicomputer during those years. His current research interests include fault-tolerant computing systems, distributed systems in the UNIX Operating Systems, multimedia operating systems, security system, and parallel algorithms.

Hyung-Bo Kim got a B.Sc. degree in Forestry and Landscape Architect and M.Sc. degree, and his Ph.D. in Urban Engineering all from Seoul National University, in 1989, 1991, and 1997. He got a certificate in MIT, Boston for Urban Studies and Traffic in 2002-2003 and he was a visiting scholar in ITS institute in UC Davis, CA in 2008. He has been Professor in the Department of Urban Engineering at Dongeui University since 1999. He is a member of KPA, KDAS and KRDA. He was a member of department of Urban Planning and Traffic System in SDI(Seoul Development Institute) and KYDI(Kyunggi Development Institute) in Seoul, South Korea. His current research interests include urban studies and traffic system, urban planning and urban design, pedestrian behavior pattern and townscape design and planning .

Sang-Bok, Yoon received a B.Sc. degree in Urban Planning and M.Sc. degree, and his Ph.D. in Urban and Traffic Engineering, Dong-A University and Waseda University(Japan), in 1987, 1989, and 1999, respectively. He is a member of KPA, KDAS and KRDA. He has been Associate Professor in the Department of Urban Engineering at Dongeui University since 2007. He was a Assistant Professor of Waseda University in Tokyo, Japan, from 1996 to 1996, and developed the Urban Traffic during those years. His current research interests include Urban Traffic systems.