PAMUKKALE ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ
PAMUKKALE UNIVERSITY ENGINEERING COLLEGE
MÜHENDİSLİK BİLİMLERİ DERGİSİ
JOURNAL OF ENGINEERING SCIENCES
YIL : 2000
CİLT : 6
SAYI : 1
SAYFA : 47-55

AN IMAGE ENHANCEMENT ENVIRONMENT DESIGNED AT 32-BIT VERSION OF VISUAL BASIC 4 PROGRAMMING LANGUAGE USING THE WIN32 API FUNCTIONS

***Aydın KIZILKAYA, *Mustafa TEMİZ, **Sezai TOKAT**
*Pamukkale University, Engineering Faculty, Electrical and Electronics Engineering Department, Denizli
**Pamukkale University, Engineering Faculty, Computer Engineering Department, Denizli

Geliş Tarihi : 08.10.1998

ABSTRACT

In this paper, using the Win32 API (Application Programming Interface) functions and MDI (Multiple Document Interface) programming technique, which is main principle of Windows system, designed an image enhancement environment at 32-bit version of Visual Basic 4 programming language is investigated. Image enhancement algorithms could be easily applied in this environment and each of results obtained could be separately showed in frames on same environment. Image enhancement techniques used in this environment are observed in spatial domain. With this program observing image enhancement techniques are contrast stretching, histogram equalization, thresholding, negative imaging, low-pass filtering, high-pass filtering and median filtering. In the filtering process of the images are utilized of the convolution techniques at this environment.

Key Words : Image enhancement, Visual basic, Win32 API functions

WIN32 API FONKSİYONLARINI KULLANARAK VISUAL BASIC 4 PROGRAMLAMA DİLİNİN 32 BİT VERSİYONUNDA TASARLANAN BİR GÖRÜNTÜ İYİLEŞTİRME ORTAMI

ÖZET

Bu çalışmada, Win32 API (Uygulama programlama arabirimi) fonksiyonları ve Windows sisteminin temel prensibi olan MDI (Çok dokümanlı arabirim) programlama tekniğini kullanarak Visual Basic 4 programlama dilinin 32-bit versiyonunda tasarlanan bir görüntü iyileştirme ortamı incelenmektedir. Bu ortamda görüntü iyileştirme algoritmaları kolaylıkla uygulanabilmekte ve elde edilen her bir sonuç aynı ortamda ayrı ayrı çerçeveler içerisinde gösterilebilmektedir. Bu ortamda kullanılan görüntü iyileştirme teknikleri; kontrast germe, histogram eşitleme, eşikleme, negatif görüntüleme, alçak geçiren filtreleme, yüksek geçiren filtreleme ve medyan filtrelemedir. Bu ortamda, görüntülerin filtrelenmesi işleminde konvolüsyon (katlama) tekniğinden yararlanılmaktadır.

Anahtar Kelimeler : Görüntü iyileştirme, Visual basic, Win32 API fonksiyonları

1. INTRODUCTION

As Visual Basic is a high level, problem oriented programming language, it is not capable of doing all of the processes related to the computer. An important thing for a programming medium is its ability to use all possibilities of the Windows system. Visual Basic programming language doesn't use all possibilities of the Windows system. It cannot be done processes such as image reading and writing directly in Visual Basic programming. Therefore, using Microsoft Windows 32-bit

Application Programming Interface, known as Win32 API, these undesirable conditions that limit performing more effective programs with Visual Basic is widely eliminated. Thus, more powerful, more functional and faster applications and implementations are designed with Visual Basic 4. For this reason, in software of image enhancement environment developed at Visual Basic programming language or these Win32 API functions have been used (Akın, 1995). The basis of designed image enhancement environment is utterly depending on using API functions. By means of these functions, images can be read and then can be saved to files after doing certain processes. Information about API is given in section 3, below.

2. GENERAL STRUCTURE OF THE IMAGE ENHANCEMENT ENVIRONMENT

Frame structure of the developed image enhancement environment has got a menu that resembles to Windows System (Figure 1).

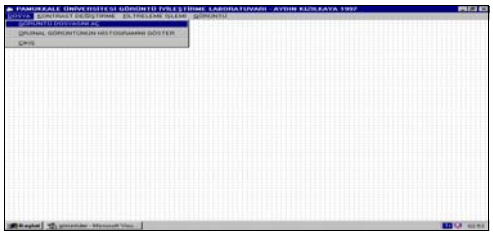


Figure 1. Menu of developed image enhancement environment

This menu environment has the ability to be modified how using these menu options requires it. By running the program, this designed menu appears on the monitor. Then, by selecting the image open file subordinate option from file option, the image open menu is shown on the monitor. This file open menu have structure and properties which are the same file open menu of the Windows System (Figure 2).

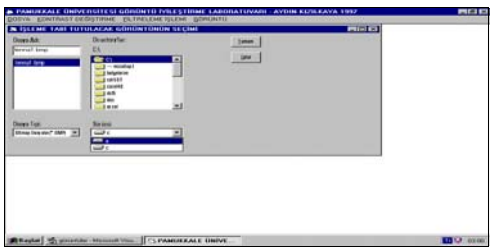


Figure 2. File open menu of the image enhancement environment

Therefore, it can easily be rearranged. Multiple-Document Interface as abbreviated MDI programming method that is the main principle of Windows System is used in this program. By using this method, subordinate frames are constituted in the main control frame. MDI allows us to have multiple forms within a single container, or "main" form. Excel, MS Word for Windows, and many other applications use the concept of MDI.

An MDI program allows us to show multiple documents at the same time, with each document in its own frame. Thus, by choosing the image enhancement options from the menu, each of image results obtained are appeared one by one on the image frame at the same environment. On the other hand, these results can be easily compared simultaneously. In this programming method each form is independent from the others. This is advantage of our program, improved at 32-bit version of Visual Basic 4 programming language (Tanyeri, 1995).

Program's skeleton consists of one MDI main form and subordinate forms related to the main form that perform various goals. Main form contains image enhancement environment menu. Connection to subordinate forms from main form is realized through short command formname.show which is written in procedures of each option. In this command, formname expression is a symbolized one form letter, it is formed whatever the name is (Kızılkaya, 1997).

In each form, there are instructions that call program class and provide to read images and write them in the determined form. By this property, there is no need to write the same procedures more than once for writing, reading and memory allocation of the images (Rimmer, 1993; Jerke and Brierly, 1996) Furthermore in each form, there are program procedures that is written to perform desired goals.

Apart from, there are dialog boxes which asks the user whether he or she wants to exit from the program or receive the numerical outputs related to image data etc. This given information processes resemble to Windows System. Moreover, different dialog boxes can easily be added to this program that will be performed to give more and assorted information according to desire.

This improved image enhancement environment has a structure that can be added a lot of image processing algorithms to this program easily. Therefore, this image enhancement environment can be transformed to an image processing environment

by adding other image processing options and procedures. Naturally, program procedures of each option must be written. Furthermore, realization of this process is easy with this improved environment (Kızılkaya, 1997). Skeleton of classes used in this image enhancement environment is given in figure 3.

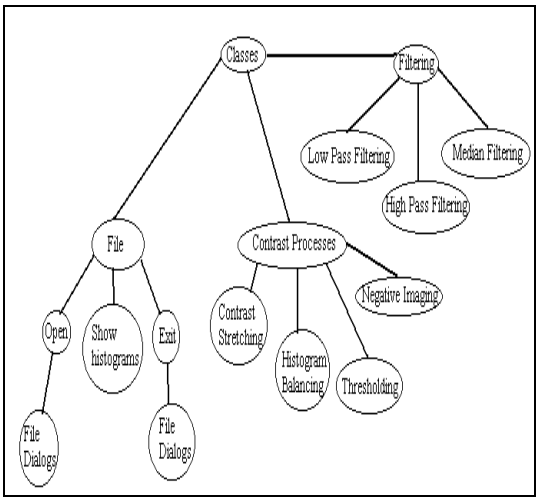


Figure 3. Structure of the image enhancement environment

3. APPLICATION PROGRAMMING INTERFACE (API)

The Windows 32-bit Application Programming Interface is a complicated set of functions, messages, and structures allowing programmers in all types of programming languages to build applications running on the Windows 95 and Windows NT operations systems. From programming languages such as C and C++, the full Windows 32API is accessible. The builders of Visual Basic had the foresight to see the language would need to be easily extended by having access to the API. They provide a balance by hiding many of the challenges of Windows programming while providing the flexibility to use the Windows environment.

The Windows 32-bit API contains more than 1,000 API calls. In addition to this, there are hundreds of windows constants, messages, and data types that go along with API and we have a complicated, yet powerful set of tools for windows programming. Furthermore, users using Visual Basic (VB) can utilize API procedures or calls to accelerate processes that VB can achieve and also operation that VB cannot accomplish.

There are many procedure libraries having prepared earlier in compiler languages. These libraries are located in EXE files of the program by linking from Windows 95 system during compiling of the designed program. Thus, program has a chance of using these procedures and functions existing in these libraries. Visual Basic as different from the other classical compilers is of link at runtime not compiling time of the program. When Windows runs a program, first it looks whether any library functions are declared or not in this program procedures. If they are done, Windows determines their address. Hence, computer knows their places when one of these functions is called in this program. Owing to this operating logic, constitution of these libraries related to procedures of each API functions written with C and Pascal programming language for obvious aims are called Dynamic Linked Libraries as abbreviated DLLs.

The most important superiority of the DLL files is that they can be changed easily and for this reason, they don't need to compile programs using themselves. Moreover, more than one application can utilize the same DLL files at the same time.

These DLL files mentioning above are libraries that contain API procedures of Windows system. API procedures are more effective than Visual Basic procedures in the point of operation performance. API calls or procedures are embedded into declarations that are part of the form or code module of Visual Basic programs before used them.

4. IMAGE ENHANCEMENT TECHNIQUES AT THIS ENVIRONMENT AND RESULTS

Investigated image enhancement techniques are contrast stretching, histogram equalization, thresholding, negative imaging, low-pass filtering, high-pass filtering and median filtering. Also, histograms of images obtained by result of some processes are important. Because, some processes which will be realized on image are done by looking at these histograms.

Histograms that show the self-curve relating to gray-level value (from 0 to 255, for image which each pixel represented with 8-bit) and quantity of pixel, can be designed at desired structure as virtually. To realize that is very easy. Different properties can be determined for these histograms by entering each option's procedure. This condition deals with

changed properties of graphic area for histograms (Kızılkaya, 1997).

4. 1. Contrast Stretching

Contrast is degree of difference between very bright and dark region in the image, as the luminance. Also, it is difference in gray level values in some particular region of an image function. High contrast images have got well appearance that forms very bright and dark regions. Dynamic range of image is part between very high and very low luminance level in the image. At the same time, it is called contrast range and is an important parameter for any imaging system (Gökmen, 1995). Before applying any process on the images, image is read to the appointed two dimensional function I(x, y) which is a function of two spatial variables through designed program. In this paper, due to studied black and white image, I(x, y) is denoted to the gray scale image's function.

Gray scale modification is a simple and effective way of modifying an image's dynamic range or contrast. With this method, the gray scale or intensity level of an input image function I(x, y) is modified according to a specific transformation. In this case, a transformation function can be determined between input image function I(x, y) and output image function g(x, y) is obtained. Also, I(x, y) and g(x, y) can be expressed as gray level values of input and output images, at (x, y) coordinates respectively. Computing the histogram of the input image and studying its characteristics can identify a good transformation in typical applications. These transformation functions may be positive or negative linear scaling function, exponential function, Root Square and parabolic curve, logarithm and inverse logarithm function. Dynamic range of the image can spread to the maximum gray level range indicated using one of these functions. This process is known as contrast stretching (Gonzales, 1981; Lim, 1990; Woods and Vernon, 1991). In gray scale digital images that each pixel is expressed with 8-bit, 2^8=256 gray levels can be mentioned. In such images, black color is symbolized by 0 gray level value and white color is symbolized by 255 gray level value.

If an image with a large dynamic range is recorded on a medium with a small dynamic range such as film or paper, the contrast and therefore the details of the image reduces, particularly in the very bright and dark regions (Lim, 1990). Also, low contrast images can result from poor illumination, lack of dynamic range in the imaging sensor, or even wrong setting of a lens aperture during image acquisition (Gonzales and Woods, 1993). Figure 4. 2 and Figure 4.3 shows such an image and its gray level histograms, respectively. At this histogram, gray levels of image have been compressed to a smaller region. Therefore, image's contrast is very low and to increase it, a function shown at Figure 4.1 is used in the designed program.

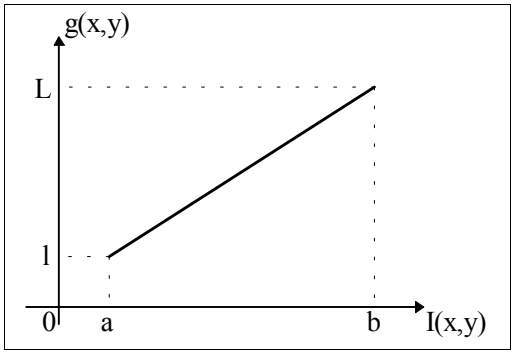


Figure 4. 1. Linear risen transformation function for contrast stretching

From the above graphic;

$$g(x,y)=\frac{(I(x,y)-a)\times L}{b-a} \qquad (1)$$

is obtained. But, the result of equation (1) may be decimal. Therefore equation (1) is represented with,

$$g(x,y)=\text{round}\left\{\frac{(I(x,y)-a)\times L}{b-a}\right\} \qquad (2)$$

Transformation function used in contrast stretching is a function of linear rising. Where (l, L) is the available gray level range and (a, b) is the gray level range of the input image. l gray level may be 0 minimum and L gray level may be 255 maximum, in images expressed with 8-bit for each pixel. Image obtained by using the equation (2) on the original image and its histogram are shown at Figure 4.4, Figure 4. 5, respectively. In Figure 4. 4 contrast stretched images are shown. Original image's gray levels are diffused between 0 to 255 gray levels as shown on the histogram (Figure 4. 5), (Gökmen 1995).

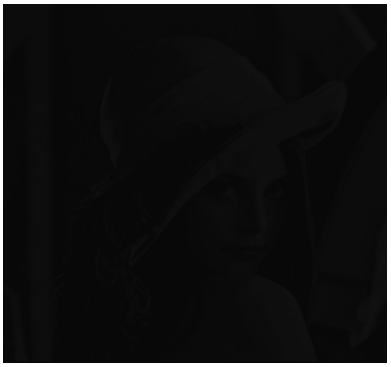


Figure 4. 2. Original image

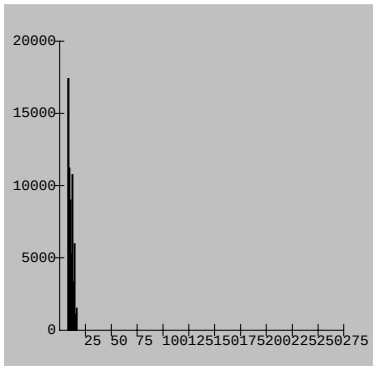


Figure 4. 3. Histogram of the original image

Figure 4. 4. Image obtained by stretching of the original image's contrast

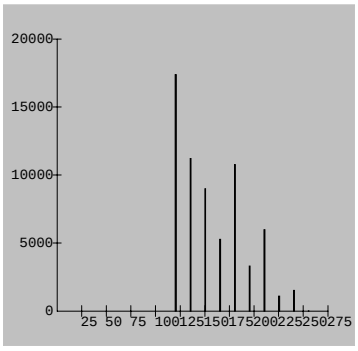


Figure 4. 5. Histogram of the contrast stretched image

Figure 4. 6. Image obtained by histogram equalizing the image at Figure 4.4

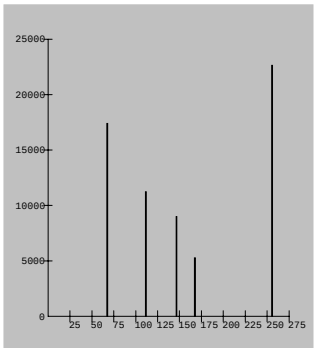


Figure 4. 7. Histogram of the histogram equalized image

4. 2. Histogram Equalization/Balancing

Gray tone adjustment in contrast stretched image may not be sufficient. Because, distribution of gray levels as quantity in the image may not be evenly. If this distribution is not balanced, image's contrast may be low or required information may remain on the dark region of the image. This undesired condition is eliminated ideally by equalizing the number of pixels per gray level. However, this cannot be achieved exactly. To obtain the function which will realize this condition is very difficult. An approximation to the function can be identified by inspection of the histogram but an analytical approach is preferable. This process is called histogram equalization or balancing. It improves contrast and intensity of images. Function using in this improved environment for histogram equalization is determined in the following, (Lim, 1984; Lim, 1990; Low, 1991)

$$F(g) = \max\left\{0, \text{round}\left(\frac{g\_levels \times t(g)}{row\,max \times col\,max}\right) - 1\right\} \qquad (3)$$

The number of pixels in a whole image is rowmax × colmax, the number of gray levels in the image (It may be 256 in 8-bit images) is g_levels, and cumulative pixel quantity is t(g). This value is obtained by adding consecutively number of pixels per gray level. At equation (3), value g is old gray level and histogram equalization function value F(g) is new gray level of the image. That is,

F(old gray level)=new gray level (4)

Also, ideal histogram would be flat with the same number of pixels at each gray level, it is given in the following:

(ideal number of pixels at each gray level)=

$$\frac{(row\,max) \times (col\,max)}{g\_levels} \qquad (5)$$

Histogram equalizing process is applied to image at figure 4.4 using the equation (3) in the developed program. Image and its histogram obtained after this process is shown at Figure 4.6 and Figure 4.7,

respectively. Clearly, the image does not contain more data as a result of histogram equalization, the data is simply presented better and the image's luminance is smoothed (Low, 1991).

4. 3.Thresholding

Thresholding is done to remove the gray level trends in an image, to make the gray level region more discrete, to split into distinct parts of an image and also to obtain two colored (black and white) images. Thresholding normally refers to setting all the gray levels below a certain level to zero; or above and equal a certain level to a maximum brightness level, (Schalkoff, 1989; Low, 1991). Maximum brightness level is 255 in 8-bit digital images. Images that have different threshold levels are shown at Figure 4.8 (a) and Figure 4.8 (b). Undesired variations in an image are eliminated through thresholding, The obtained image is a binary image. The most important thing in the thresholding process is to split the image by using a suitable threshold value. Threshold values may be determined by looking at the histogram of the original image.

Figure 4.8 (a). Original image after thresholding (Threshold value is 12)

Figure 4.8 (b). Original image after thresholding (Threshold value is 10)

4. 4. Negative Imaging

The negative of a digital image can be obtained by reverse scaling of the gray levels. Transformation function utilized in the developed program to realize this aim is shown at Figure 4. 9. In this program, the value of L is taken as 255. Digital negative images are useful in displaying medical images and in producing negative prints of images, (Lim, 1990; Gonzales and Woods, 1993).

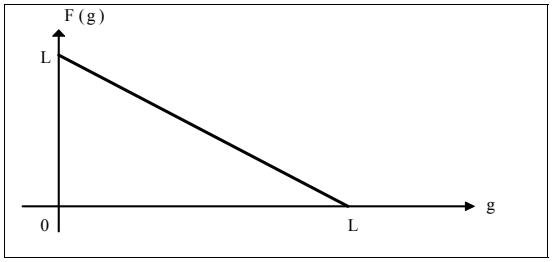


Figure 4. 9. Digital negative transformation function

Figure 4.10. Orijinal image

Figure 4.11. Its negative

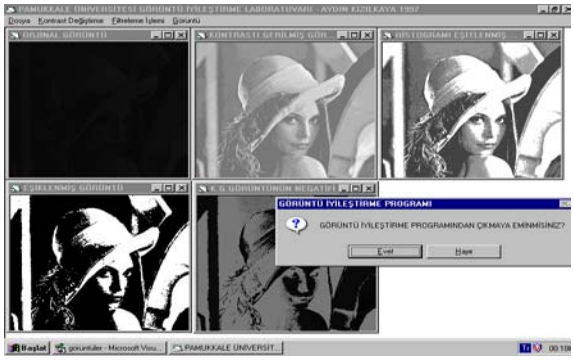


Figure 4.12. Results of image enhancement techniques, at the developed environment.

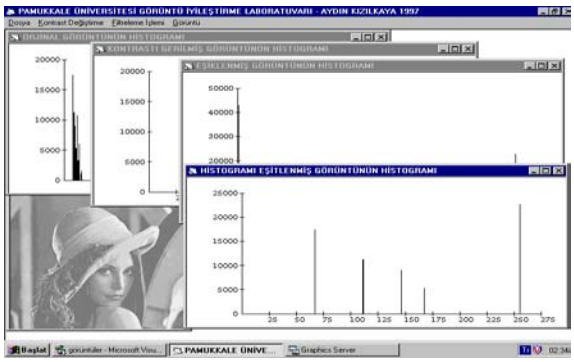


Figure 4.13. Some of the image histograms at the developed environment

4. 5. Filtering

An area process uses neighborhood information to modify pixel values or assert the existence of some property at the image points. Area process is typically used for spatial filtering. This process is utilized for removing noise, blurring or sharpening and smoothing the image and to provide this, convolution technique is utilized at this environment (Vernon, 1991; Gonzales Woods, 1993).

4. 5. 1. Low-pass Filtering

Low-pass filtering process is used to reduce the noise at the image. To apply this process, template filter is used in the spatial domain, image function I(x, y) is convoluted with template filter $h_1(i, j)$ (Hall and Awtrey, 1980; Lim, 1990; Low, 1991; Gonzales and Woods, 1993; Gökmen, 1995;). Convolution process is shown below;

$$g(x,y) = \sum_{-1}^{1} I(x+i, y+j) \times h_1(i,j) \qquad (6)$$

Template filter used in this program for low-pass filtering is,

$h_1(i,j) = 1/9 \times$

1	1	1
1	1	1
1	1	1

As shown in Figure 4.14 and Figure 4.15, noise at the original image is reduced, but blurring occurs.

Figure 4.14. Original image

Figure 4.15. Low-pass filtered image with template $h_1(i, j)$

4. 5. 2. High-pass Filtering

High-pass filtering emphasizes the edges and details of an image. It reduces the blurring and sharpens the image (Hall Awtrey, 1980; Lim, 1990; Low, 1991; Gonzales and Woods, 1993; Gökmen, 1995). Template filter used in this program for high-pass filtering is $h_2(i,j)$, and it is shown below;

$h_2(i,j) = 1/7 \times$

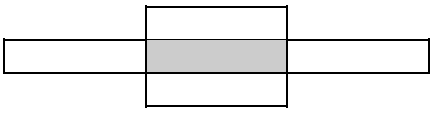
-1	-2	-1
-2	19	-2
-1	-2	-1

As shown in Figure 4.16 and Figure 4.17, blurring at the original image is reduced and edges are emphasized, but background noise increases.

4. 5. 3. Median Filtering

With median filtering, noise can be removed while preserving edges in an image. Median filtering is a popular low-pass filtering, attempting to remove noisy pixels while keeping the edges intact, (Lim, 1984; Schalkoff, 1989; Low, 1991; Gökmen, 1995). Template filter used in this program for median filtering is $h_3(i,j)$, and it is shown below;

$h_3(i,j)$ =



In median filtering, the value of the pixel being processed is replaced by the median intensity value of the pixels within the above template filter (Kızılkaya, 1997).

Figure 4.16 Original image

Figure 4. 17. Highpass filtered image with template $h_2(i, j)$

Figure 4.18. Original image

Figure 4. 19. Median filtered image with template $h_3(i,j)$

5. CONCLUSIONS

In this project, image enhancement environment that has a user-friendly menu and a modular structure has been developed. MDI programming technique and API functions are used in this developed program. Therefore, users will be able to compare each one of the image frames simultaneously. This environment can be easily transformed to image processing environment by adding image process procedures and it can be arranged again.

On the other hand, there are some disadvantages of this program. Firstly, some operations as well as histogram equalization last longer than the other programs written with C++. Secondly, image operations in frequency domain could not be done effectively due to the fact that Visual Basic 4.0 Builder does not contain concept of complex numbers. In spite of all of these limitations, it is shown that, image enhancement environment which is mostly designed using C, a medium level programming language, can be designed a more easy programming logic with an intermediate level of programming knowledge using Visual Basic, a high level programming language.

Consequently, more advanced and modular software relating to image processing applications can be realized by using Win32 API functions at advanced version of Visual Basic programming language.

6. REFERENCES

Akın, C. 1995 Microsoft Visual Basic 3.0 Professional Edition, Evrim Yayınevi, Istanbul.

Gonzales, R. C., Woods, R. E. 1993. Digital Image Processing, Addison-Wesley Publishing Company, USA.

Gökmen, M. 1995. Image Processing Lecture notes, İTÜ, İstanbul.

Hall, J. E., Awtrey, J. D. 1980. Real-Time Image enhancement using 3 × 3 pixel neighborhood operator functions, Opt. Eng.,Vol. 19, pp 421-424.

Jerke, N., Brierly, E. 1996. Visual Basic 4 API How-To, Waite Group Press, USA.

Kızılkaya, A. 1997. Taranmış Bozuk Görüntülerin Kalitelerinin Yazılım Teknikleri İle Artırılması. Yüksek Lisans Tezi, Pamukkale Üniversitesi Fen Bilimleri Enstitüsü, Denizli, Türkiye.

Lim, J. S. 1984. Digital Image Processing Techniques, Academic Press, New York, USA.

Lim, J. S. 1990 Two-Dimensional Signal and Image Processing, Prentice Hall Signal Processing Series, Englewood Cliffs, New Jersey, USA.

Low, A. 1991. Introductory Computer Vision and Image Processing, McGraw-Hill Book, England.

Rimmer, S. 1993 Windows Bitmapped Graphics, Windcrest / McGraw- Hill, USA.

Schalkoff, R. J. 1989. Digital Image Processing and Computer Vision, John Willey and Sons, Canada.

Tanyeri, M. F. 1995. İleri Düzey Visual Basic, Enternasyonel Kitabevi, İzmir.

Vernon, D. 1991. Machine Vision: Automated Visual Inspection and Robot Vision, Prentice Hall, New York, USA.

Woods, R. E., Gonzales, R. C. 1981. Real-Time Digital Image Enhancement. Proc. IEEE, Vol. 69 pp 643-654.