

*Kragujevac J. Math.* 32 (2009) 97–107.

## IMAGE PRE-PROCESSING TOOL

Olga Miljković

*College of Computer Science, Megatrend University of Belgrade,  
11000 Novi Beograd, Serbia*  
(e-mails: omiljkovic@megatrend.edu.rs, olgamkg@sbb.co.yu)

*(Received October 01, 2006)*

**Abstract.** Image pre-processing is the name for operations on images at the lowest level of abstraction whose aim is an improvement of the image data that suppress undesired distortions or enhances some image features important for further processing. It does not increase image information content. Its methods use the considerable redundancy in images. Neighboring pixels corresponding to one object in real images have the same or similar brightness value and if a distorted pixel can be picked out from the image, it can be restored as an average value of neighboring pixels. Image pre-processing tool, created in MatLab, realizes many brightness transformations and local pre-processing methods.

### 1. INTRODUCTION

Image pre-processing is the term for operations on images at the lowest level of abstraction. These operations do not increase image information content but they decrease it if entropy is an information measure [1, 2]. The aim of pre-processing is an improvement of the image data that suppresses undesired distortions or enhances some image features relevant for further processing and analysis task. Image pre-processing use the redundancy in images. Neighboring pixels corresponding to one real object have the same or similar brightness value. If a distorted pixel can be

picked out from the image, it can be restorted as an average value of neighboring pixels. Image pre-processing methods can be classified into categories according to the size of the pixel neighborhood that is used for the calculation of a new pixel brightness. In this paper, it will be presented some pixel brightness transformations and local pre-processing methods realized in MatLab [3].

## 2. IMAGE CROPPING AND FILTERING

The first step in image pre-processing is image cropping. Some irrelevant parts of the image can be removed and the image region of interest is focused. This tool provides a user with the size information of the cropped image. MatLab function for image cropping realizes this operation interactively waiting for an user to specify the crop rectangle with the mouse and operates on the current axes. The output image is of the same class as the input image.

The two-dimensional convolution operation is fundamental to the analysis of images. A new value is ascribed to a given pixel based on the evaluation of a weighted average of pixel values in a  $k \times k$  neighborhood of the central pixel. Convolution kernel or the filter mask is represented with weights supplied in a square matrix. It is applied to each pixel in an image. Discrete form of the 2D convolution operator is defined by the following relationship between the elements  $f_i(x, y)$  of the input image, the elements  $h(\alpha, \beta)$  of the convolution kernel, and the elements  $g(x, y)$  of the output image by the following master formula

$$g(x, y) = \sum_{\alpha=-(k-1)/2}^{(k-1)/2} \sum_{\beta=-(k-1)/2}^{(k-1)/2} f_i(\alpha, \beta)h(x - \alpha, y - \beta),$$

where  $x, y, \alpha$  and  $\beta$  are integers [4]. Coefficients of the kernel  $H$  represent a discrete approximation of the analytical form of the response function characterizing the desired filter. In practical cases, the kernel is a square array and  $k_x = k_y = k$ , where  $k$  is odd and much smaller than the linear image dimension. There is the following steps, realized for each pixel  $P$  represented by  $(x, y)$ :

- placement of  $H$  on  $P$ ;
- multiplication of each pixel in the  $k \times k$  neighborhood by the appropriate filter mask;
- summation of all products;
- placement of the normalized sum into position  $P$  of the output image.

This tool for pre-processing lets an user explore 2-D Finite Impulse Response filters. By changing the cut-off frequency and filter order, the user can design filter and can see the designed filter's coefficients and frequency response.

Median filtering is a non-linear smoothing method that reduces the blurring of edges and significantly eliminates impulse noise [3, 4]. It suppresses image noise without reducing the image sharpness and can be applied iteratively. The brightness value of the current pixel in the image is replaced by the median brightness of either 3-by-3 or 4-by-4 neighborhood.

### 3. INTENSITY ADJUSTEMENT AND HISTOGRAM EQUALIZATION

A gray-scale transformation  $T$  of the original brightness  $p$  from scale  $[p_0, p_k]$  into brightness  $q$  from a new scale  $[q_0, q_k]$  is given by  $q = T(p)$ . It does not depend on the position of the pixel in the image. Values below  $p_0$  and above  $p_k$  are clipped. Values below  $p_0$  map to  $q_0$ , and those above  $p_k$  map to  $q_k$ . Alpha argument specifies the shape of the curve describing the relationship between the values in the input image and output image. If alpha is less than 1, the mapping is weighted toward brighter output values. If alpha is greater than 1, the mapping is weighted toward lower darker output values. If the argument is omitted its default value is 1. Graphical controls enable an user to increase and decrease the brightness, contrast and alpha correction [5].

Another offered possibility to enhance the contrast of image, by transforming the values in an intensity image so that the histogram of the output image matches

a specified histogram, is histogram equalization technique. Region description is based on its statistical gray-level properties. Histogram provides the frequency of the brightness value in the image [4]. An image with  $n$  gray levels is represented with one-dimensional array with  $n$  elements. The  $n$ -th element of array contains the number of pixels whose gray level is  $n$ .

Assume that the pixel values are normalized and lie in the range  $[0, 1]$ . Let  $s = T(r)$ , for any  $r \in [0, 1]$ , is transformation function which satisfies the following conditions:

- $T(r)$  is single valued and monotonically increasing in the interval  $[0, 1]$ ;
- $0 \leq T(r) \leq 1$  for any  $r \in [0, 1]$ .

The original and transformed gray levels can be characterized by their probability density functions [6]. Contrast is the local change in brightness and is defined as the ratio between average brightness of an object and the background brightness. Histogram equalization technique is based on modifying the appearance of an image by controlling the probability density function of its gray levels by the transformation function  $T(r)$ . This technique enhances the contrast in the image.

#### 4. BRIGHTNESS THRESHOLDING

Brightness thresholding is an indispensable step in extracting pertinent information. A gray-scale image often contains only two level of significant information: the foreground level constituting objects of interest and the background level against which the foreground is discriminated [1]. A complete segmentation of an image  $R$  is a finite set of regions  $R_1, R_2, \dots, R_m$ ,

$$R = \bigcup_{i=1}^m R_i, \quad R_i \cap R_j = \emptyset \quad i \neq j.$$

If  $R_b$  is a background in the image, then  $\bigcup_{i=1, i \neq b}^m R_i$  is considered the object and  $R_b^C = \bigcup_{i=1, i \neq b}^m R_i$ , where  $R_b^C$  is the set complement. While there are two principal

peaks of the foreground and the background intensities in the image histogram, there are many other gray intensities present. Binarization can be accomplished by the choice of an intensity, between the two histogram peaks, that is the threshold between all background intensities below and all foreground intensities above. The input image  $I_1$  is being transformed to an output binary segmented image  $I_2$ , in the following way

$$I_2(i, j) = \begin{cases} 1; & I_1(i, j) \geq T \\ 0; & I_1(i, j) < T \end{cases}$$

where  $T$  is the threshold.  $I_2(i, j) = 1$  for the object elements and  $I_2(i, j) = 0$  for the background elements. There are different approaches for image binarization depending on the type of image [7, 8]. Successful threshold segmentation depends on the threshold selection.

A number of conditions like poor image contrast or spatial nonuniformities in background intensity can make difficult to resolve foreground from background. These cases require user interaction for specifying the desired object and its distinguishing intensity features.

## 5. CLEARING AREAS OF A BINARY IMAGE

If there is a deformation of the expected shape and size of the border and the whole region during the separation of image object from its background, it can be partially overcome. Usually small polygon mask, located next to the region and out of it, is added to clear image area with similar brightness of the region. This mask can reshape image objects and provides a separation image objects from each other and from their image background.

This operation is realized interactively, adding vertices to the polygon. Selecting a final vertex of the polygon over a white colored image region, the fill is started and recolored to black. Created fill is a logical mask and the input image is logical matrix. Using logical operator AND under logical arguments, the output image is also obtained as a logical array. If the white regions represents image foreground on the black background, whole objects or their parts can be deleted. Special user

requirements about the size and shape of the observed object, can be realized by the same way.

## 6. DETECTING EDGES

Edges are pixels where the intensity image function changes abruptly. Edge detectors are collection of local image pre-processing methods used to locate changes in the brightness function. An image function depends on two variables, co-ordinates in the image plane. Operators describing edges are expressed by partial derivatives. A change of the image function can be described by a gradient that points in the direction of the largest growth of the image function.

An edge is a vector variable with two components, magnitude and direction. The edge magnitude is the magnitude of the gradient. The edge direction is rotated with respect to the gradient direction by  $-\pi/2$ . The gradient direction gives the direction of maximum growth of the function, e.g., from black to white. The boundary and its parts are perpendicular to the direction of the gradient [2, 9]. The gradient magnitude

$$|\text{grad } g(x, y)| = \sqrt{\left(\frac{\partial g}{\partial x}\right)^2 + \left(\frac{\partial g}{\partial y}\right)^2}$$

and gradient direction

$$\varphi = \text{arg}\left(\frac{\partial g}{\partial x}, \frac{\partial g}{\partial y}\right)$$

are continuous image functions where  $\text{arg}(x, y)$  is the angle from  $x$  axis to the point  $(x, y)$ .

A digital image is discrete in nature and these equations must be approximated by differences of the image  $g$ , in the vertical direction for fixed  $i$  and in the horizontal direction for fixed  $j$ , by following equations

$$\begin{aligned}\Delta_i g(i, j) &= g(i, j) - g(i - n, j) \\ \Delta_j g(i, j) &= g(i, j) - g(i, j - n),\end{aligned}$$

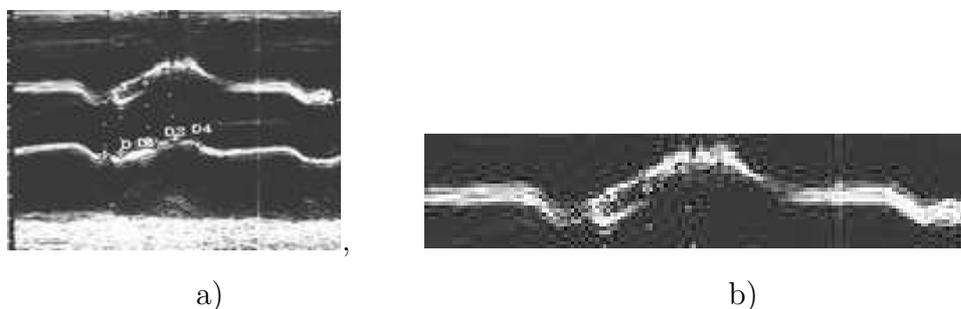
where  $n$  is a small integer chosen so to provide a good approximation to the derivative and to neglect unimportant changes in the image function.

Gradient operators approximating derivatives of the image function using differences use one or several convolution masks. Beside them, this tool uses operators based on the zero-crossings of the image function second derivative. Sobel, Prewitt, and Roberts methods find edges by thresholding the gradient and by them horizontal edges, vertical edges or both can be detected. The Laplacian of Gaussian method thresholds the slope of the zero crossings after filtering the image with a LoG filter [3]. Canny method thresholds the gradient using the derivative of a Gaussian filter.

One option in the main menu provides processing of the image region of interest or the whole image. It has to move the pointer over the image on the left and when the cursor changes to a crosshair, it has to click on points in the image to select vertices of the region of interest. Offered operations to perform are unsharpening, histogram equalization technique, lowpass filtering, median filtering, brightening, darkening, increasing contrast, decreasing contrast and boundary interpolation.

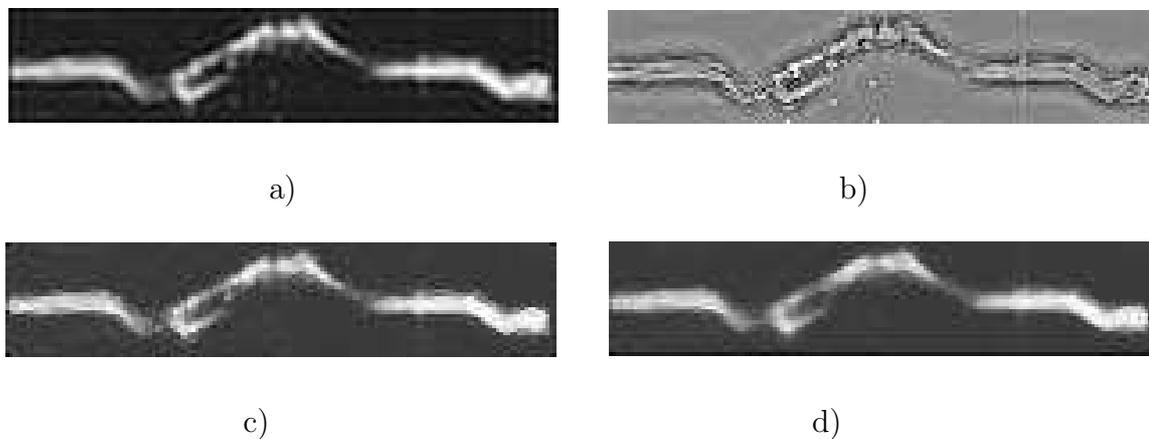
## 7. ILLUSTRATION

The tool for image pre-processing is realized in MatLab [3]. The following illustration shows the results of many pre-processing actions. The input image is doppler image of blood vessel. First, it can be cropped and its irrelevant parts can be removed (Figure 1.). Image region of interest is focused.



**Figure 1. a) Input doppler image b) Cropped image**

This image is filtered by lowpass, highpass filter and median filter method using 3-by-3 neighborhood and 4-by-4 neighborhood (Figure 2.).



**Figure 2.** a) Lowpass filtering b) Highpass filtering  
c) Median 3-by-3 filtering d) Median 4-by-4 filtering

There is a possibility of enhance the image contrast by intensity adjustment (Figure 3.) and histogram equalization technique (Figure 4.).



**Figure 3.** Intensity adjustment



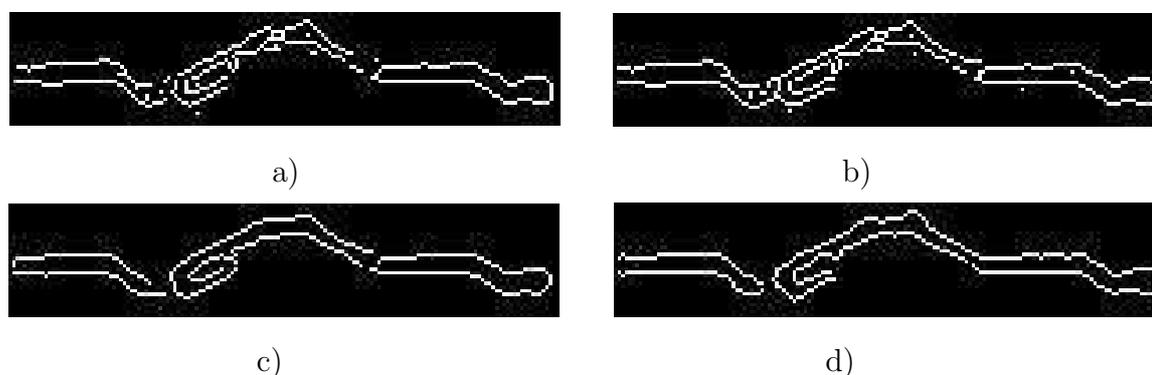
**Figure 4.** Histogram equalization

Thresholding is performed interactively. Good choice of the threshold value is essential in getting appropriate binary output (Figure 5.).



**Figure 5.** Thresholding and binary image

Results of many edge detecting methods are shown in Figure 6.



**Figure 6.** a) Sobel method   b) Prewitt method  
c) Canny method   d) Log method

## 8. CONCLUSION

Image pre-processing tool uses many useful pre-processing operations which suppress information that is no relevant to the specific image processing and enhance some image features important for further processing. They are some pixel brightness transformations and local pre-processing methods realized in MatLab.

Image cropping allows to focus attention on relevant part of an image. Image filtering use a small neighborhood of a pixel in a input image to produce a new brightness value in the output image [7]. Smoothing methods aims to reduce noise or other small fluctuations in the image. This corresponds to the suppression of high frequencies in the Fourier transform domain. Gradient operators based on local derivatives on the image function indicate locations of the image where the image function undergoes rapid changes. They suppress low frequencies in the Fourier transform domain. This software uses both techniques. Convolution-based operations are used and for edge line detectors. Operators describing edges are expressed by partial derivatives.

Gray-scale transformations modify brightness without regard to position in the image [2]. Intensity adjustment and histogram equalization enhance the contrast in the image. Brightness thresholding is a gray-scale transformation whose result is a binary image. This is a segmentation method disjointing objects and their background using a brightness constant determined interactively. Specifying the vertices

of polygon mask, it can be processed only some image regions of interest using all given operators.

Interactive work gives a chance of result improvements. User can return to the previous steps and change and correct some parameters. If there is special user requirements about a shape and size of the observed region, they can be realized clearing some region parts drawing in the vertices of a polygon fill. Impossibility of region separation from other regions by thresholding, because of prevention of border shape deformation, can be overcome using the same way. A region content can also be changed. This software is very multi-operative, flexible and gives good results. It allows further improvements and enrichments using other techniques and image pre-processing operators.

## References

- [1] Haralick, Robert M., and Linda G. Shapiro, *Computer and Robot Vision*, Volume I. Addison-Wesley, 1992.
- [2] Sonka. M, Hlavac. V, Boyle. R, *Image Processing, Analysis and Machine Vision*, PWS Publishing, 1998.
- [3] *Image Processing Toolbox User's Guide - For Use with Matlab*, Version 2, The Math Works, May 1997.
- [4] Seul. M, O'Gorman Lawrence, Sammon. Michael, *Practical Algorithms for Image Analysis - Description, Examples, and Code*, Cambridge University Press, 2000.
- [5] O. Miljkovic, M. Tuba, B. Stojanovic, *Medial-Line Detection through an Elongated Image Region*, Proceedings of III Congress of Mathematicians of Macedonia, Struga, 29.09. - 02.10, 2005, pp.503-512.
- [6] Gonzalez R., Woods R., *Digital Image Processing*, Addison-Wesley, Reading, MA, 1992.

- [7] Jain K., *Fundamentals of Digital Image Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1989.
- [8] Young I., Gerbrands J., van Vliet L., *Image Processing Fundamentals*, <http://www.ph.tn.tudelft.nl/Courses/FIP/noframes/fip-Acknowledge.html>.
- [9] Russ J., *The Image Processing Handbook*, Second Edition, CRC Press, 1994.