Use of convolution and filtration to extract information from the DCM graphic files

Mirosław Zając[1*]

[1]*Institute of Mechatronic and Information Systems,
Technical University of Lodz, Stefanowskiego18/22 90-924 Łódź, Poland*

Abstract − The paper describes the problem of using digital filters on the DICOM graphic files. An attempt has been made to improve the process of extracting information from graphic files generated by medical measuring instruments.

1 Introduction

Nowadays Computers are used in many fields of science and technology, as well as in image processing being one of such fields. We take photographs [1], tape or acquire information from medical devices and irrespective of graphic images quality all we want to get is as much information as possible [2]. The operation of extracting desired information and rejecting unnecessary data is the basis of all filtration types.

The paper presents the issue of converting and adapting the images in the form of graphic files usually generated by medical devices. The image in the file is not always adapted to our needs. By applying adequate mathematical and filtering operations it is possible to emphasize details of the image or improve its quality. The example of using the adaptation filter described in the paper indicates that sophisticated filtration techniques are effective.

---

*mirek21.mirek21@wp.pl

2   The basis of Digital Filters utilization in Image Analysis

The filters taking into account the surrounding of the analysed " pixel " [3] are believed to be one of the most useful tools in image processing.

Unfortunately, such filters perform mathematical operations on each pixel separately and hence the process is very time-consuming. The filter which takes the surrounding into account is a matrix, which by means of the coefficients included in it, describes the influence of neighbouring pixels on the analysed one. Such a matrix can be also called a filtration window or a convolution matrix because the filters perform the convolution operation. Such filters make it possible to change the " brightness" [4, 3] of the pixels according to the mathematical formula known as convolution. The mathematical formula of convolution is as follows [5, 6, 7]:

$$s(t) = \int_{-\infty}^{\infty} f(u)g(t-u)du. \tag{1}$$

For discrete signals, this formula can take the following form [5, 8]:

$$s(t) = \sum_{n} f(n)g(m-n). \tag{2}$$

The signal after the convolution has a different size from that of the input signals. For example, if the first input signal has the size N and the second has the size M, the output signal has the size $N + M - 1$. The convolution can be performed in one or two dimensions, indicated by the abbreviations 1D and 2D respectively. For the one-dimensional case [9]:

$$x[n] * h[n] = y[n] \tag{3}$$

where x and h are the input signals (for programming, they can be vectors, tables, containers), y defines the output signal, and n is the position in the signal. This means that $x[n]$ is the n-th sample in the input signal $x[n]$.

The convolution fulfils the superposition principle which means that the convolution operation is linear. Consequently, for any numbers a and b, the following equation is fulfilled [10, 7]:

$$(ax_1[n] + bx_2[n]) * h[n] = ax_1[n] * h[n] + bx_2[n] * h[n]. \tag{4}$$

For convolution of two-dimensional signals, we have [10]:

$$C[m,n] = a[m,n] * h[m,n] = \sum_{j} \sum_{k} h[j,k]a[m-j,n-k]. \tag{5}$$

For two dimensions, the position is determined by two variables. In such case, the convolution is simply the multiplication and subsequent summing of the values from both input signals.

A discrete convolution can be also written as the Cauchy product c_n of two sequences a_n and b_n [11]:

$$c_n = \sum_{k=0}^{n} a_k b_{n-k}. \tag{6}$$

Each pixel receives a new brightness value based on the brightness of the surrounding pixels. Depending on the applied parameters, this operation allows various filtration (image blurring and sharpening is possible). The pixel surrounding and the method of its processing are determined by using the filter mask, by means of applying suitable weights of these pixels. Next, the brightness of all surrounding pixels is multiplied by their weights and summed, and later it is divided by the scale parameter. The offset parameter is added to the final result. Hence, the new value of each pixel is calculated as follows:

$$J = \frac{\Sigma}{Scale} + Offset. \tag{7}$$

The new pixel value is calculated on the basis of coefficients in the filter matrix and information on the neighbouring pixels. The 2D convolution is simply an extension of the 1D convolution in two directions, instead of one.

An example of such filter is a 3x3 matrix [**12**]

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{8}$$

A similar effect can be obtained with a 5×5 matrix

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{9}$$

The only difference is that this matrix is larger and consequently more operations must be performed on the image pixels as the analysed surrounding is broader and a larger number of neighbouring pixels are included in the calculation of the new pixel value. It must be emphasized that this form of the matrix and the filter do not make any changes in the filtered image. Only the value of the pixel is taken into account, so the result of the conversion will include only the converted pixel.

The filter can also blur the image. A typical blur filter in the form of a 3×3 matrix is presented below [**13**]

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}. \tag{10}$$

Such a filter is of course very imperfect, but easily understandable. What does a blurred image mean? One needs to think what blurring operation means for an image. It can be explained that the neighbouring pixels are more similar and as a result, the details are less distinct. This can be achieved by averaging of the image area. To reduce the negative effects of filtration, sometimes the averaging filters use different coefficients than it would result from averaging of the pixel values inside the analysed area. The idea here is that the original pixel value L(m,n) should have a greater influence on the

final results than the neighbouring pixels. Higher gain is then applied to the selective field in the convolution matrix. An example of such a matrix is presented below [**12**]

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}. \tag{11}$$

The final effect is a reduced image deformation. Such a filter has, however, one significant defect: all surrounding pixels are treated in the same way. It is known that the filtering can be smarter and more selective.

A sharpening filter is based on increasing the contrast around the edge. It can be assumed that the edges in an image are dark pixels next to the bright ones. To increase contrast, one has to increase the brightness of the bright pixels and darken the dark ones even more. Then sample convolution filter with a 3x3 matrix will be as follows [**13**]:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}. \tag{12}$$

When such a filter is used, the brightness values of the neighbouring pixels are subtracted from each other. A simple relationship is formed: a dark pixel surrounded by bright pixels becomes darker, and vice versa, a bright pixel surrounded by dark pixels becomes brighter. The filter effectiveness can be adjusted by applying a suitable weight to the central matrix element (the larger the weight, the lesser the impact of the neighbouring pixels on the final result).

Edge detection is another problem which can be solved by filtration. A sample image fragment with edges is presented in Figure 1. The squares in the figure are the image pixels, and the numbers inside the squares are the "colour" [**4, 3, 14**] values coded according to the" RGB" [**4, 3**] standard. In Figure 1, there are pixels in two colours of the RGB standard which visually form two vertical columns (columns 1 and 2 form one colour entity, columns 3 and 4 form the other colour entity). The line dividing the vertical columns 2 and 3 in the figure represents the edge in the image.

An example of the calculation of the new RGB component values of the image pixels: The convolution operation has been performed for the sharpening filter with the matrix presented in (12) in respect to the colour component values shown in Fig. 1.

$$\begin{aligned} Result = (1,1)*0 + (1,2)*(-1) + (1,3)*0 + (2,1)*(-1) + (2,2)*5 \\ + (2,3)*(-1) + (3,1)*0 + (3,2)*(-1) + (3,3)*0. \end{aligned} \tag{13}$$

In this case, the kernel is in position $(2,2)$. A higher contrast has been achieved by changing the colour of the pixels lying on the edge border. The numbers in parentheses represent the coordinates in the filter matrix which subsequently have been replaced with the pixel colour values to receive the results for each RGB component. When the coordinates have been substituted by the pixel colour values, the result has been as
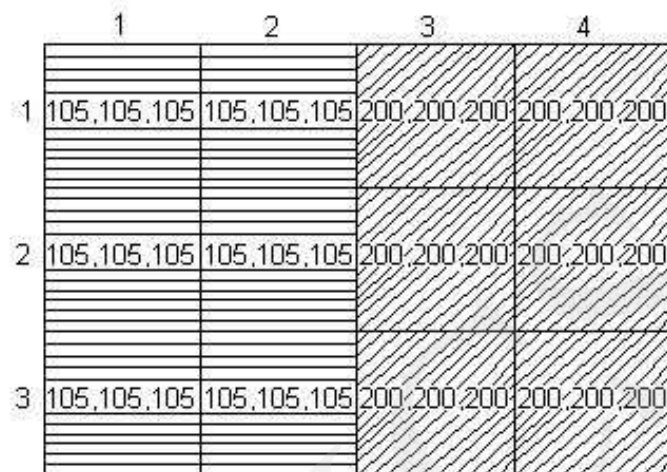
Fig. 1. A fragment of the image with marked colours (acc. to the RGB model) and edges.
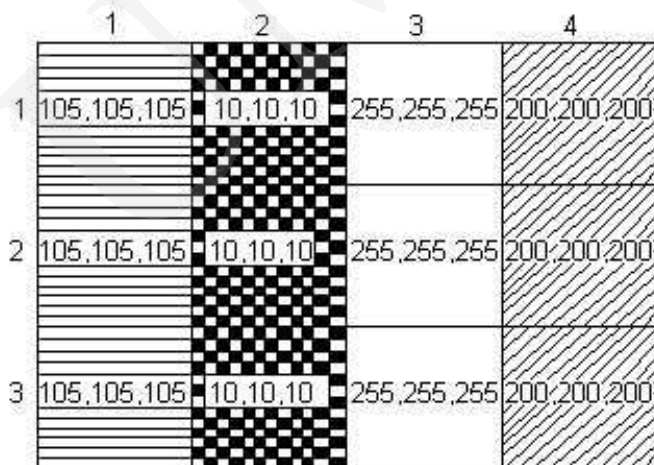

Fig. 2. An edge in the image fragment after the sharpening operation.

follows:

$$Result = 105 * 0 + 105 * (-1) + 200 * 0 + 105 * (-1) + 105 * 5$$
$$+ 200 * (-1) + 105 * 0 + 105 * (-1) + 200 * 0 = 10. \qquad (14)$$

The determined pixel will have a new colour, close to dark grey (10,10,10). Similarly, for a bright area adjacent to a dark area, the following calculation can be performed:

$$Result = 105 * 0 + 200 * (-1) + 200 * 0 + 105 * (-1) + 200 * 5$$
$$+ 200 * (-1) + 105 * 0 + 200 * (-1) + 200 * 0 = 295. \tag{15}$$

As the new colour component value is 295 and it is outside the 0-255 range specified for the 8-bit colour representation, it is changed to the highest possible value (255) which corresponds to white. Unfortunately, this method is applied on the whole area of the image, and not only on the edges, so it increases the noise and interference. It is also impossible to determine the image sharpening strength easily. The final effect of this operation is presented in Fig. 2. It is a simple contract increase which does not always yield satisfactory results. In order to obtain better results, more complicated operations and resulting selective sharpening called the USM (unsharp mask) should be used. This filter subtracts the blurred copy of the image, the unsharp mask, from the original image. The resulting copy is compared with the original and if differences are found which are greater than a predefined threshold, the images are subtracted from each other. In this filter, a threshold can be set, beginning from which the filter will act as a sharpening filter. Therefore, it is possible to choose a fragment of the image for processing and consequently, it is possible to avoid the noise amplification from a specified threshold. If the threshold is not exceeded, the image fragment will not be changed. It must be mentioned that if a fragment in the blurred version is darker than its original counterpart, the sample is brightened. If the blurred sample is darker, the procedures are the opposite. The result is a sharpened image. Usually, such filters have three settable parameters: amount, radius and threshold. Amount determines a percent value which is added at the image edges. Radius is responsible for the size of the edges to be sharpened (generally for the Gaussian blur which is used in these filters).

A small radius value results in sharpening of small details. Threshold controls change in the neighbouring tonal values for which the filter will be used. By applying a suitable threshold value one can select sharpening of distinct edges, while the softer edges will be left unchanged. Another type of filter that can be used is based on the Gaussian curve defined by the following equation [**5, 9, 15**]:

$$G(u, v) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(u^2+v^2)}{2\sigma^2}} \tag{16}$$

where u is the pixel horizontal distance from the kernel centre, v is the pixel vertical distance from the centre, and sigma is the standard deviation. The standard deviation can take any value. Based on the analyses of many images, the standard deviation equal to 3 has been applied, r is the kernel radius.

3 Using the GSM Filtration on the Computer Tomography Images

The paper presents the results of the analysis of a CT image performed by means of the Gaussian filtering. The analysis was based on the Gaussian curve defined by equation (16).

The numbers given in (17) are the filter mask weights. The radius defines the number of neighbours on each side of the source pixel which are included in the calculations. Hence, $\sigma = r/3$.

The Gaussian filtering process is performed by means of the convolution of the image function F with the mask G(u,v) which is a discrete approximation of the Gaussian function.

The filter mask weights depend on the sigma parameter (standard deviation). This parameter is responsible for image blurring - the greater it is, the larger is the smoothing effect and the noise reduction. Image blurring can be achieved by many methods, but thanks to the Gaussian distribution the kernel of this filer is circular. Shape of the blurring kernel influences the final blurring results. In the case of a circular kernel, the visual affects are the best, better than with a square kernel. The calculations were performed for the assumed radius of 3 and the results are given below.

$$
\begin{array}{ccccccc}
0.000 & 0.001 & 0.003 & 0.004 & 0.003 & 0.001 & 0.000 \\
0.003 & 0.033 & 0.147 & 0.242 & 0.147 & 0.033 & 0.003 \\
0.004 & 0.054 & 0.242 & \mathbf{0.399} & 0.242 & 0.054 & 0.004 \\
0.003 & 0.033 & 0.147 & 0.242 & 0.147 & 0.033 & 0.003 \\
0.001 & 0.007 & 0.033 & 0.054 & 0.033 & 0.007 & 0.001 \\
0.000 & 0.001 & 0.003 & 0.004 & 0.003 & 0.001 & 0.000
\end{array}
\tag{17}
$$

The central value, represented by 0.399, is typed in bold. One can easily see that the central value is the highest, the kernel is perfectly symmetrical, and the values decrease with the increasing distance from the centre.

The blur is used in a USM (unsharp mask) filter [**6, 16**]. The figure presents a half of the sharpening filter operation which first blurs the image a bit. Blurring the first half of the operation is shown in Fig. 3. The final effect is presented in Fig. 4. Fig. 5 shows the input image for the filter.

Further analysis was performed on a sample image from a "dcm" file - an original image according to the DICOM standard (Fig. 6). Unfortunately, as seen in the figure, a direct display of the image from the file does not yield the expected results (good sharpness and recognition of image fragments). All necessary information is included in the image, but it is not sufficiently legible for a human. Consequently, the image parameters must be adapted (additionally processed by filtration). After application of the colour filter, more individual elements are visible in the same image (Fig. 7).

Such filtration can often show the details which are blurred in greyscale images.
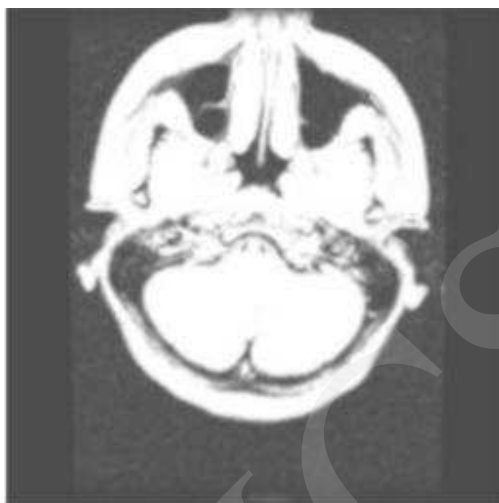
Fig. 3. Sample use of the Gaussian blur filter with the radius $r = 3$.

Fig. 4. Image without blur.

Fig. 8 shows the image after brightening and colour map conversion. Unfortunately, as we can see in the sharpened image, artefacts have appeared, i.e. the image noise has been amplified (Fig. 9). It is an undesirable phenomenon and should be avoided during the image contents analysis.

By using an adaptive filter with the coefficients given above, it was possible to sharpen the image without a noticeable degradation, achieving a so-called selective sharpening (Fig. 10). The result was an optimum image details legibility at the acceptable noise level.

Fig. 5. Sharpened image, adapted to the physical capacity of the imaging device (after conversion of the colour standard). A significant image quality improvement can be seen.
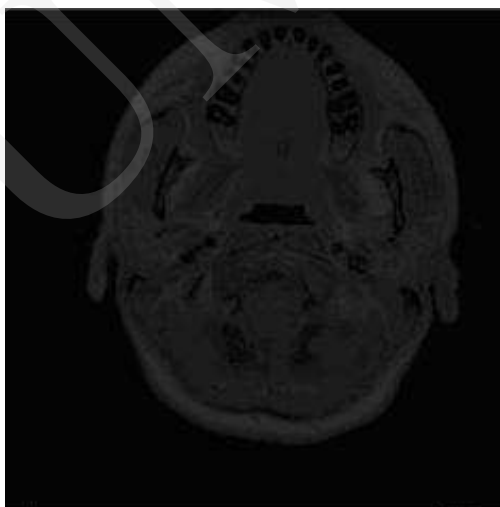

Fig. 6. Image from a dcm. file showing a head cross-section.

As shown in this example, the image process with an adaptive filter is much more legible than image sharpening with a non-adaptive filter (see comparison of images shown in Figs 10 and 9).
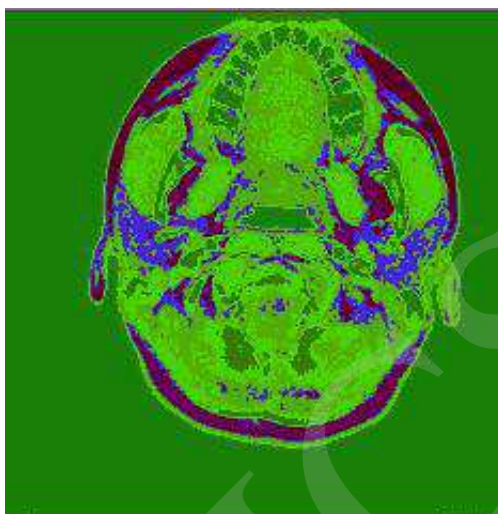
Fig. 7. Colour filtration to emphasize specified image areas.

Fig. 8. Image after brightening and colour conversion.

4   Summary and Conclusions

The paper has attempted to solve the problem related to the reading of the dcm files in the DICOM standard. The graphic files generated by measuring instruments often require additional processing to yield more information (including also legibility improvement). Such processing involves mathematical operations, filtering, and emphasizing desirable image features. Not all image quality improvement methods give
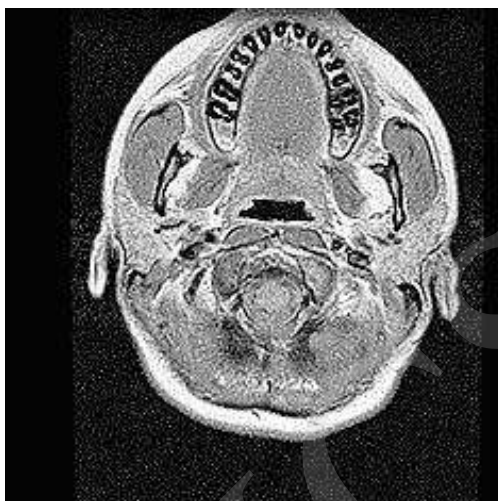
Fig. 9. Image after colour map conversion, brightening and sharpening.

Fig. 10. Final result of image processing with the adaptive filter.

acceptable results. The method applied by the author yields significantly better results (better detail legibility) than other commonly used methods.

References

[1] http://www.cabiatl.com/mricro/mricron/dcm2nii.html, sources of test images, 22.02.2011
[2] Ogiela M. R., Tadeusiewicz R., Modern Computational Intelligence Methods for the Interpretation of Medical Images, Springer (2008).
[3] Jankowski M., Elementy grafiki komputerowej, WNT (2006).

[4] Foley J.D., Wprowadzenie do grafiki komputerowej, WNT Warszawa (1995).

[5] Zieliński P. T., Cyfrowe przetwarzanie sygnałów od teorii do zastosowań, WKŁ (2005).

[6] http://homepages.inf.ed.ac.uk/rbf/HIPR2/unsharp.htm, 22.02.2011

[7] http://www.jhu.edu/signals/convolve/index.html,22,02.2011

[8] http://www.jhu.edu/signals/discreteconv2/index.html, 22.02.2011

[9] Lyons R,. Wprowadzenie do cyfrowego przetwarzania sygnałów, WKŁ (1999).

[10] http://pl.wikipedia.org/wiki/Splot_(analiza_matematyczna), 22.02.2011

[11] http://imagefilter.eu/materialy/latex/convolution_pl/convolution.pdf, 22.02.2011

[12] Tadeusiewicz R., Korohoda P., Komputerowa analiza i przetwarzanie obrazów, Wydawnictwo Fundacji Postępu Telekomunikacji (1997): 90.

[13] Korzyńska A., Przytulska M., Przetwarzanie obrazów - ćwiczenia– PJWSTK (2005): 47.

[14] http://homepages.inf.ed.ac.uk/rbf/HIPR2/quantize.htm, a color quantization 22.02.2011

[15] http://homepages.inf.ed.ac.uk/rbf/HIPR2/convolve.htm »link to Gaussian smoothing, 22.02.2011

[16] http://homepages.inf.ed.ac.uk/rbf/HIPR2/log.htm, 22.02.2011