digital image representation,
half-byte image format,
image segmentation

Zbigniew CZAPLA[1]

SEGMENTATION OF MEDICAL IMAGES IN THE HALF-BYTE DATA FORMAT

The paper presents a method of image segmentation applying the half-byte data format. The half-byte data format is an image representation that consists of two types of image pixel values, pixel base values and calculated pixel difference values. Original image pixel value is obtained by joining pixel difference value and corresponding base pixel value. The layout of base pixels value is in accordance with detected regions. Segmentation based on the half-byte image date format uses simply algorithm and is attractive computationally. The paper presents also results of segmentation medical test images converted into the half-byte data format.

1. INTRODUCTION

Contemporary medical systems utilize various imaging techniques, as: computerised axial topography imaging, magnetic resonance imaging, X-ray imaging and ultrasound imaging [4]. There exist some methods that allow defining medical parameters on the basis of image analysis.

Computer image analysis includes various techniques such as: segmentation an image into regions or objects, description of objects, classification of objects. An input of an image analysis process is analysed image and an output of this process is a set of attributes extracted from an analysed image. Image processing is often carried out before image analysis. A result of image processing can be noise reduction, contrast enhancement, image sharpening or edge detection. An input and an output of an image processing operation are images. There are known many techniques of image processing [4, 6, 10].

Segmentation is a process that merges together into regions image parts having common attributes. The basic attribute for segmentation is pixel value. The most known segmentation techniques are region growing, region splitting, boundary segmentation, texture segmentation [4, 6, 10]. There are used also other segmentation methods [5, 11, 13].

Image processing and analysis methods can include preliminary format transforms that improve process algorithm [1]. The effectiveness of image processing and analysis depends also on an applied image scanning technique that defines the input form of image data [7].

Digital images are stored in graphic files that are usually of large sizes. The size of digital image can be reduced by compression. There are many compression methods known [8, 9] which differ in complexity and effectiveness. Image compression methods often include preliminary format transforms that improve process algorithm.

The half-byte data format of digital images [2, 3] forms a new image representation. This representation is created as a result of image format conversion and consists of two types of pixel values. One type of value is a base type value, the other type of value is a difference type value that determines differences between an image pixel value and the corresponding base value. Originally pixel value is obtained by joining pixel base value and pixel difference value. The half-byte data format was originally intended to lossless image compression but this data format can be also useful to image segmentation.

2. DESCRIPTION OF THE HALF-BYTE DATA FORMAT

A digital image can be described as a two-dimensional matrix of digital values determining image pixel levels of a grey or colour component. The proposed model assumes greyscale images with an intensity resolution of 8 bits per pixel. The data representation of a greyscale image of 8 bits per pixel can

[1] Silesian University of Technology, Faculty of Transport, Department of Transport Informatics Systems, e-mail: Zbigniew.Czapla@polsl.pl

be interpreted as a one-byte data format. Image conversion into the half-byte data format constitutes a new data representation.

Some classes of images contain regions of pixels which values are similar to one another. Pixels of such a region can be described by a sequence that is composed of a base value and a set of difference values determined as differences between the current pixel value and the region base pixel value. The entire image representation includes all individual region sequences. Assuming that all pixel difference values do not exceed 15, one different value requires 4 bits for encoding and two difference values can be encoded in one byte by joining together neighbouring pairs of difference values. Base pixel values are encoded without the change as unsigned 8 bit numbers.

The half-byte data format of digital images is defined for images with spatial resolution 256×256 of pixels. The examined image is described by image matrix ***X*** containing elements $x_{m,n}$. Image vector ***Y*** and conversion vector ***H*** are created during processing.

2.1. IMAGE SCANNING

The examined image is divided into separate square blocks with resolution 2 x 2 of pixels. Each image block is described by block coordinates j (rows of blocks), k (columns of blocks) and ordinal block number N defined by

$$N = 128j + k, \qquad j = 0, 1, ..., 127, \qquad k = 0, 1, ..., 127. \tag{1}$$

All blocks of image matrix ***X*** are transformed correspondingly into image vector ***Y***. Pixels of each block are described by inner block pixel coordinates m (rows of pixels) and n (columns of pixels). The transformation of a single block with block coordinates j (rows of blocks), k (columns of pixels) and inner block pixel coordinates m (rows of pixels), n (columns of pixels) into corresponding elements of image vector ***Y*** describes the equation:

$$y_{4N+i} = x_{2j+m,\ 2k+n}\,, \qquad i = 2n + m \qquad \text{for} \quad m = 0,1, \quad n = 0,1. \tag{2}$$

All blocks of matrix ***X*** are transformed one by one into image vector ***Y.*** After conversion image vector ***Y*** contains image pixels in order determined by block number N.

2.2. IMAGE ENCODING

Image encoding begins with determination of the first base value

$$b = y_0. \tag{3}$$

A marker denoted c and a base value b split in half-byte pixel values by expressions calculated using arithmetic operators div (integer division) and mod (remainder) are written into conversion vector ***H***

$$\begin{aligned} h_0 &= c, \\ h_1 &= b \operatorname{div} 16, \\ h_2 &= b \bmod 16. \end{aligned} \tag{4}$$

Elements of image vector ***Y*** are successively read and two cases are considered for each element. In the first case the difference between the pixel value and the base value allows four-bit coding and the difference value is written down into conversion vector ***H***. Indices of the vector elements indicate current writing position, Δi determines an index offset of conversion vectors ***H*** and image vector ***Y.***

$$h_{i+\Delta i} = y_i. \tag{5}$$

In the second case the difference between the pixel value and the base value is too big to carry out four-bit coding. Then the new base value is defined

$$b = y_i \tag{6}$$

and half-byte values of marker *c* and base value *b* are written correspondingly into conversion vector ***H***

$$\begin{aligned} h_{i+\Delta i} &= c, \\ h_{i+\Delta i+1} &= b \operatorname{div} 16, \\ h_{i+\Delta i+2} &= b \bmod 16. \end{aligned} \tag{7}$$

Sub-stream ***S*** is created by joining together neighbouring difference values of conversion vector ***H***. All types of half-bytes (markers and half-bytes of pixel value) may be joined.

$$s_i = h_{2i} \cdot 16 + h_{2i+1}. \tag{8}$$

Output stream is written into target image output file. Target image output file consists image representation in the half-byte data format.

2.3. IMAGE DECODING

Image decoding is performed as back conversion from the half-byte data format into the one-byte data representation. Back conversion is carried out in reverse order and the individual steps follow as: bytes of input stream ***S*** are split into half-byte values, conversion vector ***H*** is reconstructed on the basis of half-byte values obtained from input stream ***S***, image vector ***Y*** is filled with pixel values reconstructed on the basis of elements of conversion vector ***Y***, image matrix ***X*** is filled with pixel values reconstructed on the basis of elements of image vector ***Y***.

The image obtained as the result of the reconstruction from input stream ***S*** and the source image are exactly the same.

2.4. IMAGE SEGMENTATION

Image segmentation is carried out during image back conversion from the half-byte format into the one-byte format. The segmentation process involves determining a number of segmentation base value ranges $L>1$. For each segmentation base value range ($l=1,\ldots,L$) the following parameters are defined: maximum pixel base value of the range denoted r_{lmax}, minimum pixel base value of the range denoted r_{lmin} and common base pixel value of the range denoted r_l.

In the course of decoding conversion vector ***H*** is reconstructed by splitting input stream ***S*** into half-byte values. Conversion vector ***H*** contains half-byte values: markers, half-bytes of pixel base value and difference half-bytes. Next step of back conversion is calculation of succeeded elements y_i of image vector ***Y*** using elements of conversion vector ***H***. For each element y_i to begin with pixel base value b_i is calculated (Δb determines the current index offset of pixel base value and pixel difference value)

$$b_i = h_{i-\Delta b} \cdot 16 + h_{i-\Delta b+1} \tag{9}$$

and then the final value of element y_i is composed (Δi determines an offset of indices of conversion vectors ***H*** and image vector ***Y***)

$$y_i = b_i + h_{i+\Delta i}. \tag{10}$$

The segmentation algorithm assigns pixel base value b_i of each decoded element of image vector ***Y*** to the right base value range

$$r_{l\ \min} \le b_i \le r_{l\ \max} \tag{11}$$

and replaces calculated element of image vector ***Y*** with common base pixel value of the range

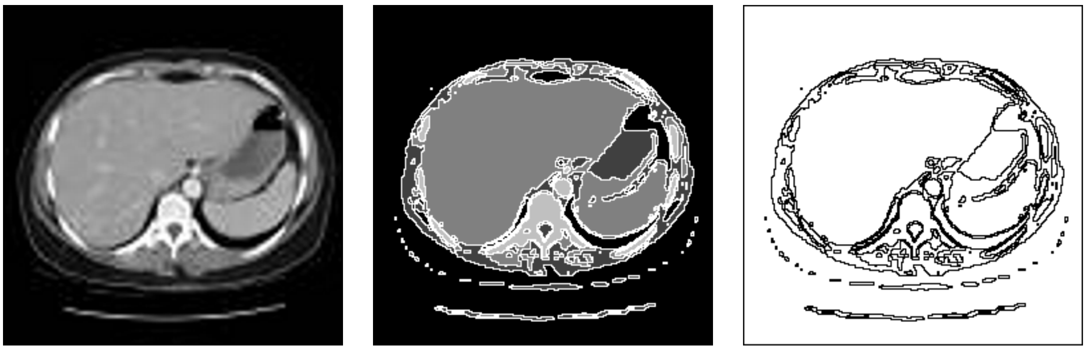
$$y_i = r_l. \tag{12}$$

Image matrix ***X*** is reconstructed on the basis of elements of image vector ***Y***. in accordance with the scanning order used in the course of coding. Segmentation regions are constitute by merging neighbouring pixels of the same range value r_l in rows and in columns.

The detail level of partitioning depends on the number and the size of segmentation ranges. The basic algorithm assumes regular division of image greyscale into disjoint segmentation base value ranges. Taking into consideration 256 grey levels the number of pixel base ranges must be multiple of two. However image greyscale can be divided into segmentation ranges of various size setting on the basis of e.g. image histogram. Thus image greyscale division corresponds to image prosperities.

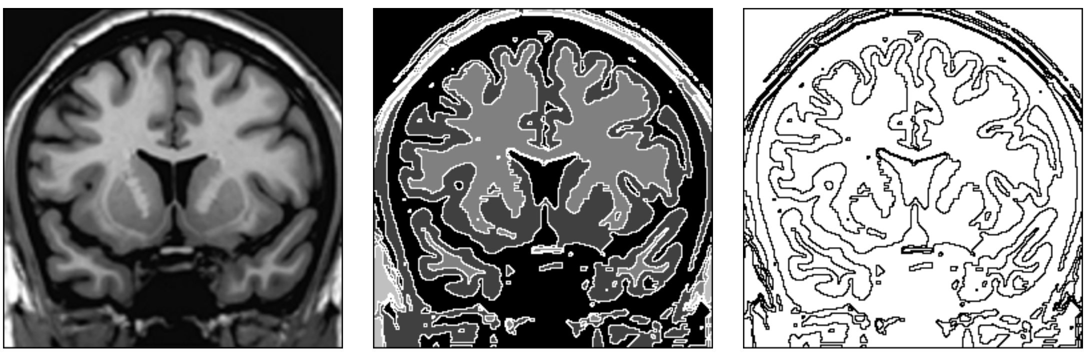
3. TEST RESULTS

Segmentation of medical images in the half-byte data format is applied to three test images [12]: Image 1, Image 2 and Image 3. All test images are in greyscale 8 bits per pixel and with resolution 256 x 256 of pixels. The number of the segmentation base value ranges is equal L=4 The original test images and the result test images are shown respectively in Figure 1 to Figure 3. The original test images are placed at the left part of figures, the result of segmentation images are placed in the centre of figures and the images showing the contour boundaries are placed at the right part of figures.



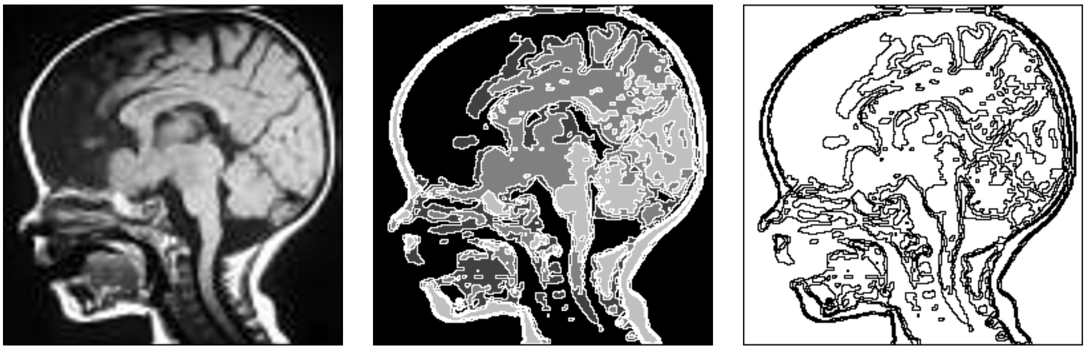
(a) (b) (c)

Fig. 1. Result of segmentation of Image 1: a) original image, b) result image, c) contour boundaries.



(a) (b) (c)

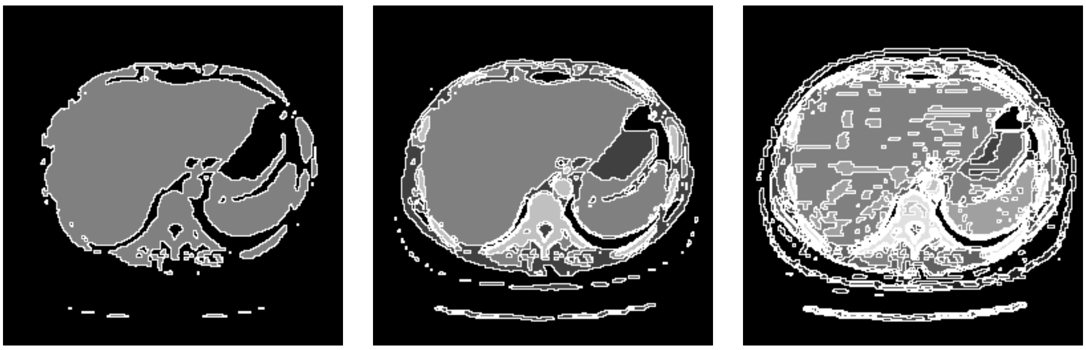
Fig. 2. Result of segmentation of Image 2: a) original image, b) result image, c) contour boundaries.



(a) (b) (c)

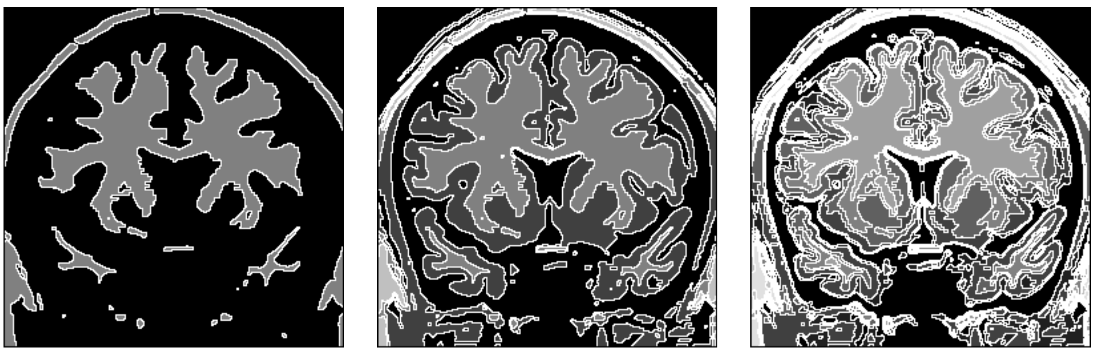
Fig. 3. Result of segmentation of Image 3: a) original image, b) result image, c) contour boundaries.

Segmentation of images in the half-byte data format allows adapting the detail level of partitioning. Figure 4 to Figure 6 show the results of segmentation depended on the number of the base value ranges.



(a) (b) (c)

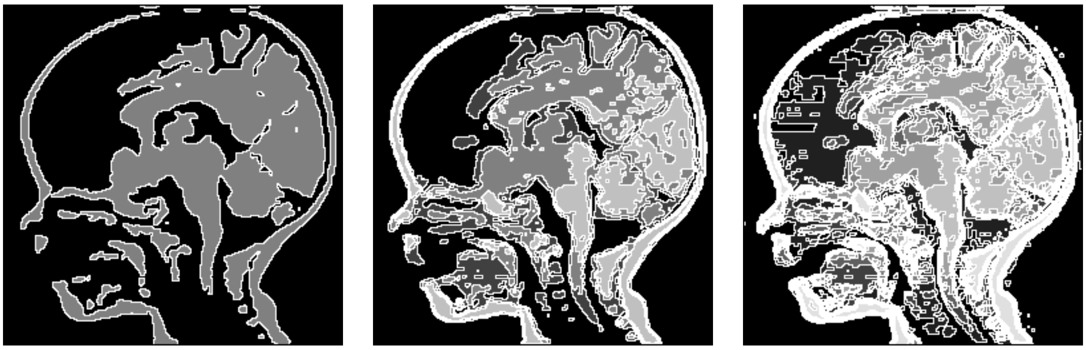
Fig. 4. Result of Image 1 segmentation using the various number of base value ranges: a) L=2, b) L=4 c) L=8.



(a) (b) (c)

Fig. 5. Result of Image 2 segmentation using the various number of base value ranges: a) L=2, b) L=4 c) L=8.

Enhancement of individual image regions can be achieved by appropriate increasing selected base value ranges.



(a) (b) (c)

Fig. 6. Result of Image 3 segmentation using the various number of base value ranges: a) L=2, b) L=4 c) L=8.

4. REFERENCE TO OTHER SEGMENTATION TECHNIQUES

Segmentation of images in the half-byte format is generally a technique using region growing. However this method utilize operations on clusters (groups) of pixels. Thus that kind of segmentation has also somewhat features of a statistical method. Criterion of pixel belonging to cluster is the same pixel base value. Performing operation on the one value for all pixels of cluster allows to reduce significantly a number of operation of segmentation algorithm. Table 1 contains the size of pixel base values for test images and also operation reduction ratio (the size of all test images is 64 KB).

Table 1. Size of base values and reduction ratio.

Test image	Base values (KB)	Reduction ratio
Image 1	15,0	0,23
Image 2	22,6	0,35
Image 3	23,6	0,37

The test images converted into the half-byte format give the size of base values in the 15,0-23,6 KB range. The operation reduction ratios are in the 0,23-0,37 range. These parameters indicate that segmentation images in the half-byte format is effective.

5. CONCLUSIONS

The half-byte image data format is suitable for segmentation. The advantage of the half-byte image data format is applying simply processing algorithm. Conversion into the half-byte image data format creates a new image representation. The layout of base pixel value is in accordance with image regions.

The half-byte image data format is intended to so call “natural” greyscale images 8 bits per pixel. Conversion into the half-byte data format allows segmentation with the use of a simple algorithm and therefore this image format is attractive computationally. Image conversion into half-byte data format can be useful also for hardware implementation of image processing algorithm. The half-byte image data format gives usually a smaller image representation than an image bitmap and can be used as a method of preliminary image data transform.

BIBLIOGRAPHY

[1] CZAPLA Z., Video Images Fast RLE Compression Methods, Monograph selected by J. Piecha: Transaction on Transport Systems Telematics. Theories and Application, pp. 88–95, Silesian University of Technology Academic Press, Gliwice, 2006.

[2] CZAPLA Z., The Half–Byte Descriptors for Image Data Units, The 12th International Conference Medical Informatics and Technologies, 15–17 November 2007, Osieczany. Journal of Medical Informatics & Technologies, Vol. 11, pp. 115–121.

[3] CZAPLA Z., Half–Byte Format of Images from a Road Traffic Data Recorder, Transport Problems, Silesian University of Technology Academic Press, Gliwice, Vol. 4, Issue 2, 2009, pp. 85–91.

[4] GONZALES R.C., WOODS R.E., Digital Image Processing, Pearson Prentice Hall, New Jersey, 2008.

[5] GRADY L., Random Walks for Image Segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 28, No. 11, 2006, pp. 1–17.

[6] PRATT W.K., Digital Image Processing, John Wiley & Sons, Inc., Hoboken, New Jersey, 2007.

[7] SKARBEK W., Metody reprezentacji obrazów cyfrowych, Akademicka Oficyna Wydawnicza PLJ, Warszawa, 1993, (in Polish).

[8] SALOMON D., Data compression, The Complete Reference, Springer–Verlag, New York Berlin Heidelberg, 2000.

[9] SAYOOD K., Introduction to Image Compression, Morgan Kaufman Publishers, San Francisco, 2000.

[10] TADEUSIEWICZ R., KOROHODA P., Komputerowa analiza i przetwarzanie obrazów, Wydawnictwo Fundacji Postępu Telekomunikacji, Kraków, 1997, (in Polish).

[11] TAO W., JIN H., ZHANG Y., Color Image Segmentation Based on Mean Shift and Normalized Cuts, IEEE Transaction on Systems, Man, and Cybernetics–Part B: Cybernetics, Vol. 37, No. 5, 2007, pp. 1382–1389.

[12] Website: www.healcentral.org (April 2010).

[13] ZOUAGUI T., BENOIT–CATTIN H., ODET C., Image segmentation function model, Pattern Recognition 37, 2004, pp. 1785–1795.