TRANSPORT PROBLEMS 2009
PROBLEMY TRANSPORTU Volume 4 Issue 2

Zbigniew CZAPLA
Silesian University of Technology, Faculty of Transport
Krasińskiego St. 8, 40-019 Katowice, Poland
Corresponding author. E-mail: Zbigniew.Czapla@polsl.pl

HALF-BYTE FORMAT OF IMAGES FROM A ROAD TRAFFIC DATA RECORDER

Summary. The paper proposes the image half-byte format applying to images from a road traffic data recorder. The format defines image pixel values using differences between a pixel value and a determined base value. The image half-byte representation can be further processed by appropriate methods. The paper presents also results of converting test images into the half-byte format.

PÓŁBAJTOWY FORMAT OBRAZÓW Z REJESTRATORA DANYCH RUCHOWYCH

Streszczenie. W artykule zaproponowano półbajtowy format obrazów zastosowany do obrazów z rejestratora danych ruchowych. Format określa wartości pikseli korzystając z różnic między wartością piksela a ustaloną wartością bazową. Półbajtowa reprezentacja obrazu może być dalej przetwarzana przy wykorzystaniu odpowiednich metod. Przedstawiono również wyniki konwersji obrazów testowych do formatu półbajtowego.

1. INTRODUCTION

Contemporary traffic control systems utilize digital images. Graphic files containing digital images have usually a large size. Size reducing of these files is possible by applying different compression methods. Effectiveness of compression is determined by the compression factor that presents the ratio of the image file size before compression and after compression.

Many compression methods are known [3, 4, 5]. The compression methods differ in complexity and effectiveness. Compression can be either lossless or lossy. Lossless compression algorithms create an image representation that allows full image reconstruction. Applying lossy compression a part of the image information is lost and this permits to achieve the significant size reduction of the image representation. Compression factors obtained as a result of lossy compression are much larger than at lossless compression. However lossy compression can cause significant deterioration in quality of reconstructed images. Lossy compression cannot be applied in some areas, everywhere where full image reconstruction is required. Data compression, in particular image data compression, has great meaning for traffic control systems, especially for the ones using data transmission over communication lines.

Image compression methods can incorporate preliminary format transforms that improve processing algorithms in operation [1]. The compression effectiveness depends also on an applied image scanning technique [6].

The image conversion into the half-byte format [2] is not a compression technique. The result of this conversion is a new image representation, which pixels are described by differences between a pixel value and a base value. A large part of these differences can be coded with 4 bits in the form of half-byte. Pairs of half-bytes are joined together into bytes. A stream of such bytes makes the image half-byte format.

2. DESCRIPTION OF THE METHOD

A digital image is a two-dimensional object that can be described as a table of values which determine shade levels of image pixels. The proposed method assumes grayscale images with 8 bits per pixel (8 bpp) resolution. The 8 bpp resolution is sufficient for the majority of technical applications. The 8 bpp grayscale image can be interpreted as an image at one-byte format. The image representation after conversion can be interpreted as an image at half-byte format.

Many of images contain regions of similar to one another in value pixels. Pixels of such regions can be described by sequences consisting of a base pixel value and a set of differences between the current pixel value and the base pixel value. It is assumed that the difference between the base value and the individual pixel value is determined by the value not exceeding 15. This permits 4 bit coding. The one-byte base pixel value is divided into two half-byte values called the more significant half-byte and the less significant half-byte. The entire image half-byte sequence is obtained by joining all individual region half-byte sequences into one half-byte sequence. The image sequence of half-bytes is converted into the image sequence of bytes by joining together following pairs of neighbouring half-bytes into one-bytes. This is carry out both for base half-bytes and difference half-bytes. The obtained byte sequence is the image at half-byte format.

2.1. Standard encoding

The examined image with the resolution of 256 x 256 pixels is described by the image matrix denoted ***X*** containing elements $x_{m,n}$. The conversion vector denoted ***Y*** and a half-byte vector denoted ***H*** are created during processing. The image is divided into separated square blocks *B* x *B* of pixels. Each block is described by block coordinates *j* (rows), *k* (columns) and ordinal block number *N* defined by

$$N = \frac{256}{B} j + k, \qquad j = 0, 1, \ldots, \frac{256}{B} - 1, \qquad k = 0, 1, \ldots, \frac{256}{B} - 1 \tag{1}$$

The example of image division into blocks 64 x 64 of pixels shows Figure 1.

	k =0	1	2	3
j =0	N=0	1	2	3
1	4	5	6	7
2	8	9	10	11
3	12	13	14	15

Fig. 1. The image division into blocks 64 x 64 of pixels
Rys. 1. Podział obrazu na bloki 64 x 64 piksele

Blocks of image matrix ***X*** are transformed into conversion vector ***Y***. Pixels of each block are described by inner block coordinates *m* (rows) and *n* (columns). The transformation of a single block with block coordinates *j, k* and inner block coordinates *m* (rows), *n* (columns) into appropriate elements of vector ***Y*** describes equation

$$y_{N \cdot R \cdot R+i} = x_{m+(j \cdot B),\ n+(k \cdot B)} \tag{2}$$

where:

$n = 0, 1, \cdots, B-1,$

$m = 0, 1, \cdots, B-1,$ for even n,

$m = B-1, B-2, \cdots, 0,$ for odd n,

$i = m + n \cdot B.$

The example of the image block conversion into conversion vector ***Y*** (the first and the second block 4 x 4 of pixels) shows Figure 2.

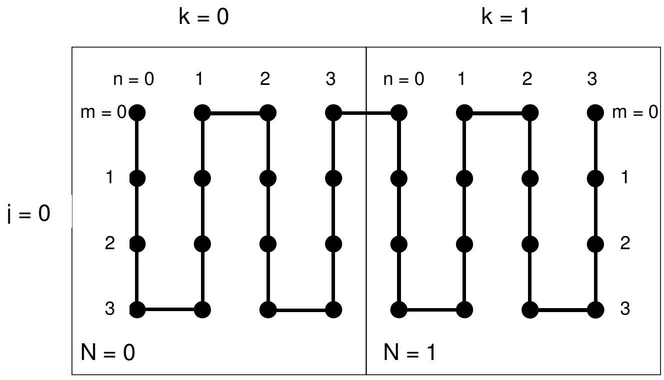

Fig. 2. The image block conversion into the conversion vector
Rys. 2. Konwersja bloków obrazu na wektor konwersji

All blocks of matrix ***X*** are transformed appropriately into conversion vector ***Y***. The next step is determining the base value

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

and calculation two first elements of half-byte vector ***H*** by expressions using arithmetic operators div (integer division) and mod (remainder)

$$\begin{aligned} h_0 &= y_0 \operatorname{div} 16 \\ h_1 &= y_0 \bmod 16 \end{aligned} \tag{4}$$

The half-byte vector ***H*** is created on the basis of conversion vector ***Y*** values. There are two cases. In the first case the differences between the pixel value and the base value permits to write down the pixel value as the half-byte value $f(y_j)$. In the second case the difference between the pixel value and the base value is too big and the marker *c* is written into half-byte vector ***H*** (indexes of the vectors appropriately indicate current writing position). The equation shows possible half-byte values written into half-byte vector ***H***.

$$h_i = \begin{cases} c & \text{marker half - byte} \\ f(y_j) & \text{pixel value half - byte} \end{cases} \tag{5}$$

If the marker half-byte is written there is necessity to set the new base value

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

and to write it into half-byte vector ***H***

$$\begin{aligned} h_{i+1} &= y_j \operatorname{div} 16 \\ h_{i+2} &= y_j \operatorname{mod} 16 \end{aligned} \tag{7}$$

The output stream ***S*** is created by successive joining together pairs of half-bytes of half-byte vector ***H***

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

All types of half-bytes may be joined. Output stream of the bytes (joined half-bytes) is written into the image output file.

2.2. Dynamic encoding

Standard encoding assumes the permanent base value for all pixels of the region. Dynamic encoding applies the individual base value for each pixel of the image. The base value is equal to the value of the previous pixel and is determined by expression

$$b_j = y_{j-1} \tag{9}$$

The whole pixel value is written into half-byte vector ***H*** only in cases when the difference between the pixel value an the base value is too big to write down the pixel value as a half-byte value. The whole pixel value is divided into two half-bytes. Two half-bytes of whole pixel value are preceded by the marker and the sequence of three half-bytes is written into half-byte vector ***H***

$$\begin{aligned} h_i &= c \\ h_{i+1} &= y_j \operatorname{div} 16 \\ h_{i+2} &= y_j \operatorname{mod} 16 \end{aligned} \tag{10}$$

Dynamic determining of the base values permits to create longer sequences of half-byte values between whole pixel values written in half-byte vector ***H***.

2.3. Decoding

Decoding of the image at half-byte format has to be adapted to a kind of encoding and it is performed in reverse order to encoding. The individual steps follows as:

- the input stream ***S*** is split in half-bytes and half-byte vector ***H*** is appropriately written,
- conversion vector ***Y*** is filled with pixel values reconstructed on the basis of the values of half-byte vector ***H***,
- image matrix ***X*** is filled with pixel values reconstructed on the basis of the values of conversion vector ***Y***.

The image obtained as a result of decoding of the half byte sequence and the source image are the same. Conversion of most images into the half-byte format gives a smaller representation than the original image. The half-byte format can be regarded, in such cases, as a form of lossless compression.

3. TEST RESULTS

The conversion into the half-byte format is applied to four test images: Image 1, Image 2, Image 3 and Image 4. All of four test images are in grayscale with 8 bits per pixel (8 bpp) and with the resolution 256 x 256 of pixels. The test images are shown appropriately in Figure l to Figure 4 on the left side of the figures. All of four test images are converted into the half-byte format. The images at the half-byte format are placed on the right side of the figures.

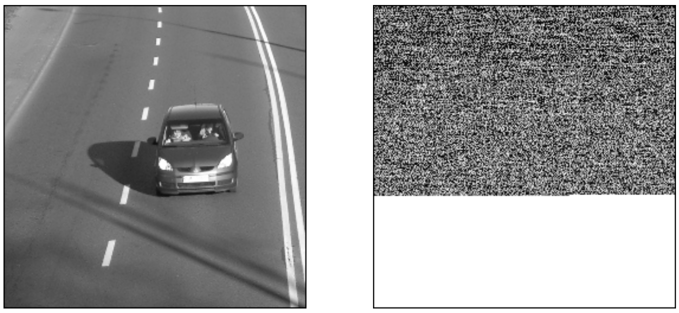

Fig. 3. Image 1 and its half-byte format
Rys. 3. Obraz 1 i jego format półbajtowy

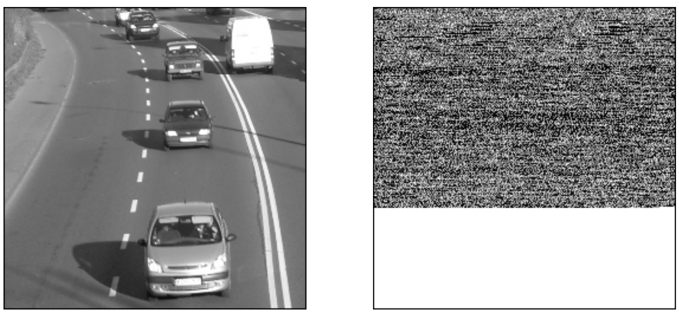

Fig. 4. Image 2 and its half-byte format
Rys. 4. Obraz 2 i jego format półbajtowy

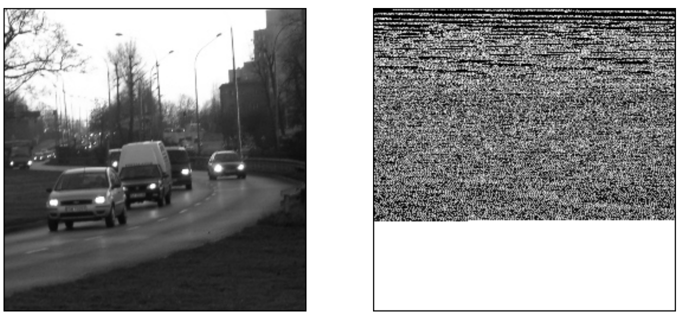

Fig. 5. Image 3 and its half-byte format
Rys. 5. Obraz 3 i jego format półbajtowy

Fig. 6. Image 4 and its half-byte format
Rys. 6. Obraz 4 i jego format półbajtowy

All of test images are at BMP format thus the size of the test images is 66,6 kilobytes. The smallest size of output file is obtained using blocks 2 x 2 of pixels. Table 1 contains the sizes of output half-byte files of the test images. The conversion is carried out using blocks 2 x 2 of pixels both for standard half-byte encoding and dynamic half-byte encoding. Table 1 contains also, for comparison, the sizes of output files at GIF format which was obtained by dictionary compression of the same test images.

Table 1

The output file sizes (in kilobytes) of the test images

Test image	**Standard half-byte encoding**	**Dynamic half-byte encoding**	**GIF format**
Image 1	42,6	41,1	53,5
Image 2	45,0	43,5	54,7
Image 3	47,3	45,9	52,5
Image 4	52,0	51,0	57,3

Compression factors are calculated for all of test images. The results from Table 1 are the basis for determining of compression factors. Table 2 contains the compression factors of the test images. The compression factors are computed for the half-byte format (standard encoding and dynamic encoding) and GIF format.

Table 2

The compression factors of the test images

Test image	**Standard half-byte encoding**	**Dynamic half-byte encoding**	**GIF format**
Image 1	1,56	1,62	1,25
Image 2	1,48	1,53	1,22
Image 3	1,42	1,45	1,27
Image 4	1,28	1,31	1,16

The half-byte format gives the size of test image output files in the range 42,6-52,0 for standard encoding and 41,1-51,0 for dynamic encoding. The size of test image output files at the GIF format are in the range 52,5-57,3. The compression factors are at the half-byte format for standard encoding in the range 1,28-1,56, for dynamic encoding in the range 1,31-1,62 and at the GIF format in the range

1,16-1,27. The half-byte format is more effective than GIF format both for standard encoding and dynamic encoding.

The half-byte format improves the compression effectiveness of so called “nature” images, which give usually poor compression results. The image at the half-byte format can be further compressed using appropriate compression algorithms. The main advantage of the half-byte format is that the images at such format can be more effectively compressed.

4. CONCLUSIONS

The image half-byte conversion is a method of preliminary transformation and it improves the effectiveness of compression. The half-byte format is intended for so called “natural” images that give poor results of lossless compression. The half-byte format assumes grayscale 8 bpp images however, it can be also apply to colour images taking into consideration that each of the image colour components is processed separately.

The half-byte format of images improves compression efficiency with the use of simple algorithms. The half-byte image format can be useful for hardware implementation of image processing algorithms and also for parallel processing.

References

1. Czapla Z.: *Video Images Fast RLE Compression Methods. Monograph selected by J. Piecha: Transaction on Transport Systems Telematics. Theories and Application.* Wydawnictwo Politechniki Śląskiej, Gliwice, 2006, pp. 88-95.
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. Drozdek A.: *Wprowadzenie do kompresji danych*. Wydawnictwo Naukowo-Techniczne, Warszawa, 1999.
4. Salomon D.: *Data compression. The Complete Reference*. Springer-Verlag, New York, Berlin Heidelberg, 2000.
5. Sayood K.: *Kompresja danych. Wprowadzenie*. Wydawnictwo Read Me, Warszawa, 2002.
6. Skarbek W.: *Metody reprezentacji obrazów cyfrowych*. Akademicka Oficyna Wydawnicza PLJ, Warszawa, 1993.

Received 11.12.2008; accepted in revised form 15.06.2009