High-speed Methods of Image Compression/Decompression

D. LAZAR, I. CIMPOIES, Gh. SOROKIN, S. GRITCOV
Technical University of Moldova
gritscov@gmail.com

***Abstract* –This thesis deals with the study of methods of images compression / decompression that require less hardware and time resources, compared to the classical methods. The results of the implementation of these image processing techniques in Delphi 7 are presented. There was carried out a comparative analysis of different methods of compression / decompression.**

***Index Terms* — compression, time resources, the quality of the image.**

I. INTRODUCTION

Image compression methods used in modern systems are mainly based on JPEG methods, if the image is static, and - on MPEG, if the picture is dynamic. In the systems required high-speed image processing, such as image compression, transmitted from the satellite to the Earth in real time, the application of the JPEG method is unacceptable. In this case, are considered methods which can reduce the time of image processing in tens times. Some definite changes in the JPEG method reduce the time of image processing significantly. A different approach in image processing is compression based on the correlation properties of the image. These two methods will be discussed in this article.

II. COMPRESSION BASED ON DISCRETE COSINE TRANSFORM

Discrete cosine transform (DCT) is applied in the compression algorithm with losses, such as JPEG and MPEG. By means of DCT, JPEG compression algorithm transforms the image representation in time domain to its spectral interpretation. In the matrix of the coefficients, obtained after DCT, low frequency components are concentrated in the upper left corner, while the high frequency components - at the right bottom corner. Because of small amplitude of the images high frequency components, they can be neglected, equating to zero. As a result, the coefficient matrix contains only low-frequency components, which reduces the total number of important coefficients and can significantly reduce the size of the image compared to the original one.

DCT is based on the multiplication of three matrices:

$$M = T * A * T', \quad (1)$$

where M - the matrix after the transformation, A - pixel values, T - transformation matrix, T '- transposed matrix of the transformation. The decrease of the transformation matrix size reduces the time spent on DCT.

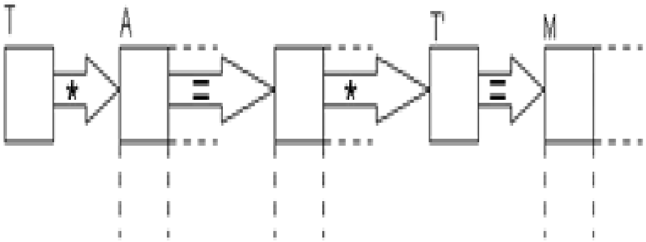

Fig. 1. Schematic representation of the process of DCT.

Below is the code that performs matrix multiplication.

```
for w := 0 to Round(ImW/MD) - 1 do
            //ImW – image width (in pixels),
            //MD – transformation matrix size
 for h := 0 to  Round(ImH/MD) - 1 do
            //ImH – image height (in pixels)
  for i := 0 to MD - 1  do
   for j := 0 to MD - 1 do
   begin
    b[w*MD + i, h*MD + j] := 0;    //b – the result of
                                   //multiplying
     for k := 0 to MD - 1 do
      b [w*MD + i, h*MD + j] := Round(b [w*MD + i,
h*MD + j] + (C[i,k] * PrI[w*MD + k, h*MD + j]));
   end;
```

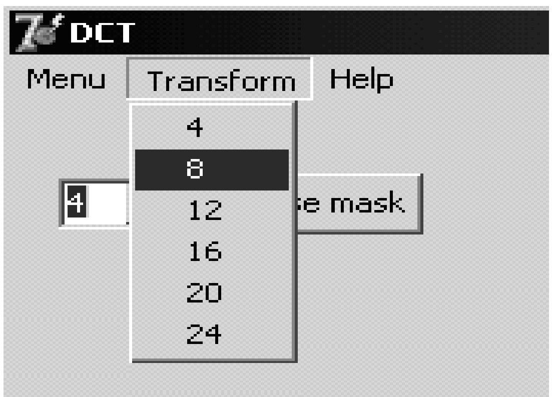

Fig. 2. The interface of designed software

To estimate the time of image processing and the degree of compression was designed software, performing DCT. The interface of designed software is shown in Fig. 2.

.

a)

b)

c)

Fig. 3. An example of the original image (a), compressed in 2 times (b) and compressed in 4 times (c).

In the experiments were obtained certain results. For example, the compression of the 4 Mbytes image with a compression ratio equal to 2 and the size of the transformation matrix 4 * 4 runs for 44,040,192 cycles, which at a frequency of the processor, for example, at 200 MHz, and taking into account that each command is executed in one cycle, the compression time is 0.22 seconds.

Under the same experimental parameters as in the first example, but with the size of the transformation matrix 8 * 8 the compression is being performed in over 94,371,840 cycles, which takes 0.44 seconds.

III. THE COMPRESSION METHOD BASED ON THE CORRELATION PROPERTIES OF THE IMAGE

Most of the images have strong correlation properties. Thus, the values of nearby bytes, which carry information about the image, differ slightly - within a few minor bits that allows to equate this bytes. Therefore, it is possible to transfer not all the bytes, but only the first one and the number of similarities with him. This method, which is called the compression method based on the correlation properties of the image, can significantly reduce the size of the original image.

Let’s consider the processing of images in the BMP format by this method. The BMP format image can be represented as a two-dimensional array, in which the each element is a pixel, which contains information about the corresponding color.

$$\begin{pmatrix} r_{11}, g_{11}, b_{11} & r_{12}, g_{12}, b_{12} & \dots \\ r_{21}, g_{21}, b_{21} & r_{22}, g_{22}, b_{22} & \dots \\ \dots & \dots & \dots \end{pmatrix}$$

Fig. 4. Representation of the BMP image, where r - the red pixel, g - green pixel, b - blue pixel.

From this array are allocated 3 bytes for each color, which are read in line. Next is a comparison of each byte with the first up to the k most significant bits. When they are equal in the new array is written the byte value and the number of bytes. For example: $\left(r_1 \quad n_1 \quad r_7 \quad n_2 \quad r_8 \quad r_3 \quad r_{20} \quad n_4 \quad \dots \quad \dots\right)$ where ri - byte value, ni - the amount of bytes, and so is for each of the three colors (red, green, and blue).

The inverse transformation is performed so: instead of byte ni, which displays the amount of repetitive bytes, in the matrix representing the final image, is being written byte ri ni times, and so - for each of the three colors.

a)

b)

Fig. 5. Representation of the BMP image (a) and compressed image (b).

a)

b)

Fig. 6. Representation of the BMP image (a) and compressed image (b).

The method was implemented in Borland Delphi. Software algorithm consists of 3-parts: setting initial values, image processing and displaying the results on the screen. An excerpt from the program of processing a single byte in red is shown below:

```
pixel:= Image1.Canvas.Pixels[u,v];
        // Copying of the current pixel in the variable
        // pixel
r:=GetRValue (pixel);
        // Allocation of the red component
if ( (r1 and mask) = (r and mask) )and(nr<255)  then
      // Masking the bit, not involved into the
      comparison   //and the comparison with the
      previous byte
```

```
begin
    nr:=nr+1;
// if the bytes are similar, there is an increment of
// the counter
  end
 else
// otherwise
  begin
// in the current cell of the being created array
    red[ir]:=r1;
// is being written the first similar byte,
    red[ir+1]:=nr;
// and in the next one - the number of bytes
    ir:=ir+2;
// increment of the counter array
    r1:= GetRValue (pixel);
// updating the value that will be compared with
    nr:=0;
// reseting the counter
  end;
  Image2.Canvas.Pixels[u,v]:=RGB(r1,g1,b1);
// image restoration
```

The compression degree depends on the image size and its type. The result of the program is shown in Fig. 5-6. In both cases, the compression was based on the first 5 bits. However, in the first case (Fig. 5), the compression factor was 2.6, in the second - 1.5 times, because the last image contains small details. Thus, after the processing with the greater accuracy the resulting image size can be larger than the original one.

The degree of deterioration in quality also varies under different conditions. Thus, implementation of this method on programmable logic, takes 0.146s at 100 MHz clock frequency for processing of the 4MB image.

IV. ANALYSIS OF THE RESULTS

Analyzing the considered methods of compression/decompression it can be concluded that the choice of a particular method of compression is determined by various conditions. If it is known priory what kind of images are processed (for example, all images contain only large objects and etc.), then the preferred compression method is the one based on the correlation properties of the image, which requires n times less time for image processing. If the images which are processed are unknown, or the dimensions of objects in the image are changing, the more acceptable method is DCT (Discrete Cosine Transform), which can provide the same level of image compression, regardless of the processed image type.

V. CONCLUSION

Implementation of DCT can significantly reduce the time of image processing, in comparison with the JPEG method. The reducing of the image processing time by 4 times in comparison with the method of DCT can be achieved by application of the method based on the correlation properties of the image. Disadvantage of this method is the necessity to know the type of processed images. These methods can be applied in satellite communication systems of image transfer in real time, as they require much less hardware and time resources in the implementation of compression / decompression, in comparison with classical methods.

REFERENCES

[1] Безруков В. Н, *Разработка и применение элементов теории преобразования сигналов изображений в системах прикладного телевидения.* Автореферат диссертации на соискание ученой степени доктора технических наук. М. – 1995 г.

[2] Р. Гонсалес, Р. Вудс, *Цифровая обработка изображений.* Техносфера, Москва, 2005 г., 1072 с.

[3] Рабинер Л., Гоулд Б., *Теория и применение обработки сигналов.* Мир, Москва, 1978 г., 848 с.

[4] Tinku Acharia, Ping-Sing Tsai, *JPEG2000 standard for image compression: concepts, algorithms and VLSI architectures.* Published by John Wiley & Sons, Inc., Hoboken, New Jersey, 2004, 296 p.

[5] Блаттер К, *Вейвлет-анализ.Основы теории.* Техносфера, Москва, 2006 г., 279 с.

[6] Сэломон Д., *Сжатие данных, изображений и звука.* Техносфера, Москва, 2004, 368 с.

[7] Прэтт У., *Цифровая обработка изображений*: Пер. с англ. Мир, Москва, 1982.— Кн.1 — 312 с.

[8] Фурман Я. А., Юрьев А. Н., Яншин В. В., *Цифровые методы обработки и распознавания бинарных изображений*. Изд-во Краснояр. ун-та. Красноярск, 1992 г., 248 с.

[9] Majid Rabbani, Paul W. Jones, *Digital Image Compression Techniques*. SPIE Press, 1991, 221 p.

[10] Weidong Kou, *Digital Image Compression: Algorithms and Standards*. Springer, 1995, 192 p.