COLOR IMAGE COMPRESSION USING DISCRETE COSINUS TRANSFORM (DCT)

Adietiya R. Saputra

Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Gunadarma
Jl. Margonda Raya no. 100, Depok 16424, Jawa Barat
adietiya.rs@gmail.com

Abstract

Multimedia continues to grow. One of the most widely used elements in multimedia is images. An important component of the image is color. In the development of multimedia, color image is more often used because they are more interesting and informative. Along with these developments, a way to store and send images efficiently is required with the color quality of the image is maintained. The purpose of this research is to compress the color in color images so that the image is more efficient in storage, but people can still capture the information provided by that image. In this research 2D-DCT (Discrete Cosine Transform) is used for compressing a color image in order to have less memory but still have a good quality. The result is a compressed image with much smaller size than the source image but does not look much different when viewed by the human eye. The result showed the percentage of compression achieved 95.88 percent for extreme image and achieved 97.59% for the soft color (homogeneous). This technique does not damage the image and has a high compression ratio for lossy compression.

Keywords : *image compression, DCT, quantization*

INTRODUCTIONS

Currently, multimedia is growing rapidly and has become an inseparable part of human life. Multimedia is a tool that can create dynamic and interactive presentation that combines text, graphics, animation, audio and video [1]. An important component in multimedia is the image. An image provides visual information that can be captured by the human eye. The information has been widely used in various fields such as entertainment, design, photography, publishing and promotion even in the field of defense and mapping.

In the development of multimedia, color image is more often used because they are more interesting and informative. A grayscale image represented by 8 bits contains 256 Kbyte of data. With color information, the number will be tripled. Images have different characteristics so that an image compression method best done to an image do not certainly suitable on other image. An important component of the image is color. Color component can provide visual information that can be quickly accepted by the human eye. By looking at the color, a person is able to identify or capture the information contained in an image. One of the important things in color compression is the color difference between the original image with decompression image is can not be seen by the human eye when there is no comparison image, so that information can still be accepted. It is expected that the proposed method will maximize the compression of color image. Compression can be either lossy or lossless. Lossless compression reduces bits by identifying and eliminating statistical redundancy. No information is lost in lossless compression [2].

Lossy compression reduces bits by identifying unnecessary information and removing it [3].The process of reducing the size of a data file is popularly referred to as data compression, although its formal name is source coding (coding done at the source of the data before it is stored or trans-mitted) [4].

Image data comprise of a significant portion of the multimedia data and they occupy the major portion of the communication bandwidth for multimedia communication [5]. Previous research on image compression is growing rapidly. DCT is one of the transforms used for lossy image compression such as the JPEG image [6]. The purpose of this research is to compress the color in color images so that the image is more efficient in storage, but people can still capture the information provided by that image.

RESEARCH METHOD

In this research, the author uses DCT transform to compress color image and earnings compression ratio from reconstructed image. The complete block is shown in Figure 1 below.

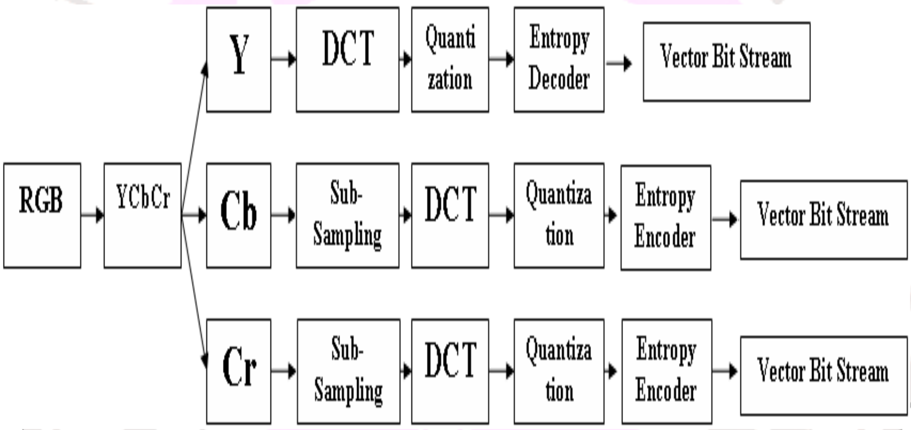


Figure 1. Block diagram of compression process using the DCT

Figure 2. Image Source Peppers.bmp

The source image are three color planes of 8-bits per-pixel each representing Red, Blue and Green (RGB). These are the colors used by hardware to generate images and for this research and for this research we used the Peppers. bmp image as an image sourceThe first step is to convert the source image from RGB , to ycbcr color space, the aim to separate the Y or luminance component represent the brightness and chrominance represent color information. The image source are display in Figure 2.

Reasons for converting to ycbcr is that the luminance component is more important than the chrominance components on human perception. After that, do subsampling of Cb and Cr compo-nents by a factor 2 in ieach dimension. This is first lossy component.and gives a factor of 2 compression.

The next step is to partition each component images into 8x8 blocks. Each of these blocks is then process separately by the DCT transformation. This is needed because the DCT has no locality information whatsoever. There is only frequency information available. The division of image into 8x8 blocks is a tradeoff between compression, image quality and complexity. smaller blocks is simpler to implement but less for compression. Larger blocks more complicated but there are more frequency coefficients. Each pixel value in the 2-D matrix is quantized using 8 bits that produce value in the range 0 to 255. All values are level shifted by subtracting 128 for each value to the -128 to + 127 before computing DCT because DCT are designned to work on the pixels between -128 to+127. This makes the values symmetric around zero with average close to zero.

DCT-II applied to each row and column of each block. The result is a transformation coefficient of 8 x 8 in which the element (0,0) (left-top) is the DC component (zero frequency) and other data on the vertical and horizontal index greater represent the horizontal and vertical spatial frequency is higher. So far this does not introduce any loss or compression.

During quantization, the size of DC and AC coefficient is reduced. Quantization is done by dividing the output DCT process with a specified value in quantization matrix. After quantization, most of the high frequency coefficient (the lower right corner) is zero. To exploit the number of zeros, a zig-zag scan of the matrix is used. This method has the advantage that low-frequency coeffients are placed first, while high frequency components come last in the array.

Zigzag positioning of quantized transform coefficients can be seen in the Figure 3 below :

1	2	6	7	15	16	28	29
3	5	8	14	17	27	30	43
4	9	13	18	26	31	42	44
10	12	19	25	32	41	45	54
11	20	24	33	40	46	53	55
21	23	34	39	47	52	56	61
22	35	38	48	51	57	60	62
36	37	49	50	58	59	63	64

Figure 3 : Zig-zag positioning of quantized transform coefficients

The final step of compression process is enthropy encoding. This encoding will eliminate a number of zeros so it will be less data in the form of a vector of bits.

Decompression Process

The process of reconstructing the image is just the invers of the compression process with the same steps. The first process is the entropy decoder and DeZigzag scan. Decoding functions to compile back into a bit vector matrix. This process will return the number of zeros that have been removed in the process of encoding and organizing them into their original position as it was before encoded.

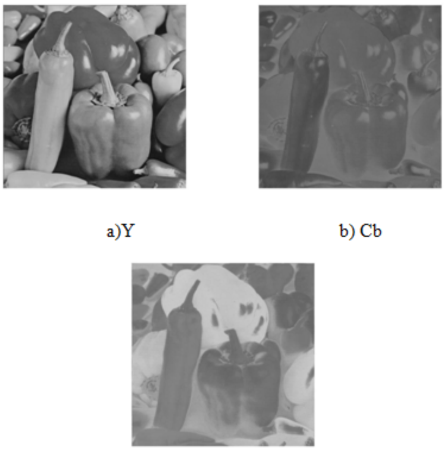
The De-Zigzag scan is the inverse process of Zigzag scan. The coefficients are put back to their position before zigzag scanning. After that, De Quatization process will restore the DCT coefficient matrix by multiplying the matrix with luminance and chrominance quantization table. And then DCT coefficient matrix of this process will be retur-ned to the value of pixels making up the image with inverse DCT, after inverse DCT, each component of image adds 128 for returning the original pixel value. After that, Cb and Cr components are returned to their original size, that is the same size with the Y component, this is necessary so that the three components can be combined to form a YCbCr image. The last process is to return the YCbCr color space to RGB color space.

RESULT AND DISCUSSION

One of the image used in this research is peppers.bmp with a size of 768 Kb. It can be seen in the Figure 4 below. In this research we used Matlab to run the compression method. The first stage is to prepare the image to be compressed. The YCbCr image of Peppers can be seen in the Figure 5 below. Results Separation of each component YCbCr can be seen in the Figure 6 below :

Figure 4. Image Source Pepper.bmp

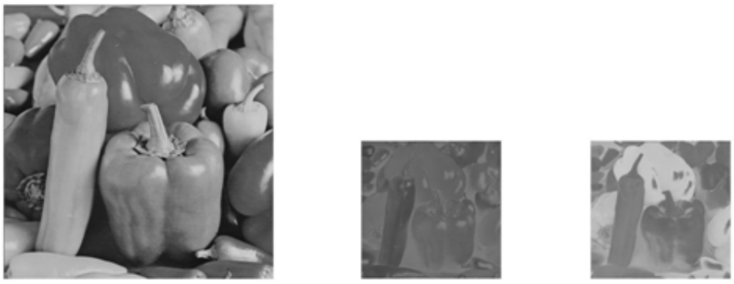
Figure 5. YCbCr image of Peppers



a)Y b) Cb

c)Cr

Figure 6. Image Component Y, Cb and Cr



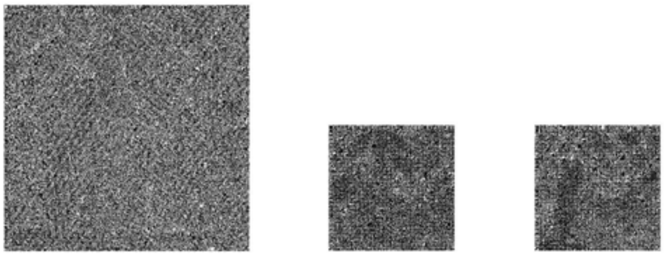
a)Y b)Cb c)Cr

Figure 7. Component Y and Cb and Cr which have been in SubSampling

a)Y b)Cb c)Cr

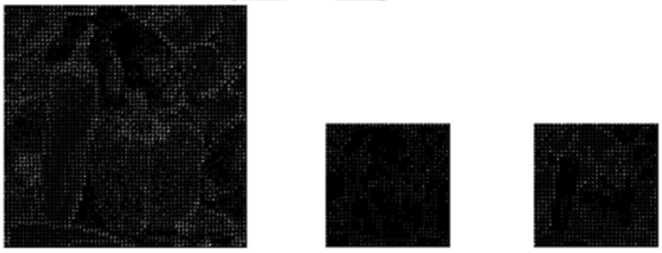
Figure 8. Image components Y, Cb and Cr substracting by 128

After separated, Cb and Cr components in SubSampling. Each component of separated image become as in Figure 7. After the image is divided into 3 elements. Each image element must be subtracted by 128 before DCT . The result after substracting can be seen in the Figure 8. After substracting 128, the image has pixel values ranged between -128 - 127 and is ready to be transformed. DCT is performed on each sub image 8x8 from the image. The image is still composed of pixel values. These values must be transformed with the DCT before the Quantized. Matrix sub image 8x8 of component Y before the transformation. And the results of DCT Transform can be seen in the Figure 9.



a)Y b)Cb c)Cr

Figure 9. DCT coefficients of the components Y, Cb and Cr



a)Y b)Cb c)Cr

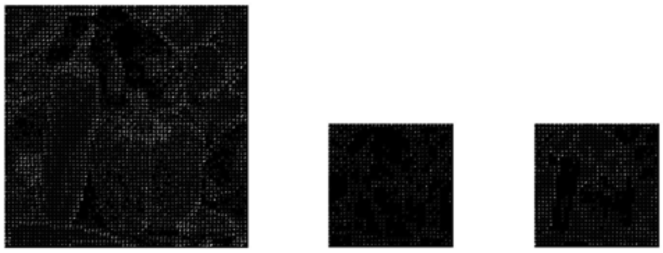
Figure 10 DCT coefficient quantization results from component Y, Cb and Cr

After transformed, the image is not identified because now it no longer contains the image pixel values, but consisted of frequencies that contain the image. After DCT, next step is quantization. At this stage, there is lossy compression where bits that will represent the colors contained in the image are reduced. Quantization results are shown in Figure 10.

In the sample 8x8 matrix above shows that most of the value of AC of the matrix has a value of zero. After quantization is done, next entrophy encoding. To do this, first do zigzag scanning of bits quantization results. Zigzag scanning is done separately by column where each column consists of 64 rows. The result is a column vector with EOB (the highest score) the back of each nonzero final value in each column. EOB value, replaces a number of zeros after the last nonzero in each column. Vector length of each component Y, Cb and Cr will vary depending on the number of zero value. Stages of entropy coding is the latest stage in the process of compression. For the next will be decompressed to get the image reconstruction.

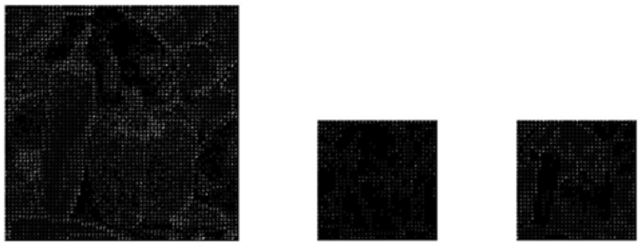
Decompression

Decompressing the image of just doing the opposite process of compression. The first process is entropy decoder then followed de quantization , inverse DCT, adding 128 on each bit images and combine the back of each element of the image into YCbCr image and convert it into RGB image. Doing Decoding encoding for each component of the image. The principle is just the opposite of the entropy decoder and the zigzag scan, re-encoded bit vectors into the matrix. The coefficients are put back to Their corresponding original positions. Decoding results of Sub Image 8x8 Y component. The results of the above program is a matrix with 64 columns, similar to the matrix before zigzag. After that, the matrix is returned to its original matrix. The result for all elements can be seen in the in Figure 11.



a)Y b)Cb c)Cr

Figure 11 Component Y, Cb and Cr after decoding

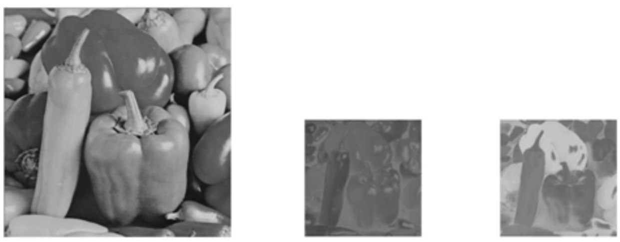


a)Y b)Cb c)Cr

Figure 12. Coefficient deQuantization of component Y, Cb and Cr

a)Y b)Cb c)Cr

Figure 13. Result invers dct to component Y, Cb and Cr



a)Y b)Cb c)Cr

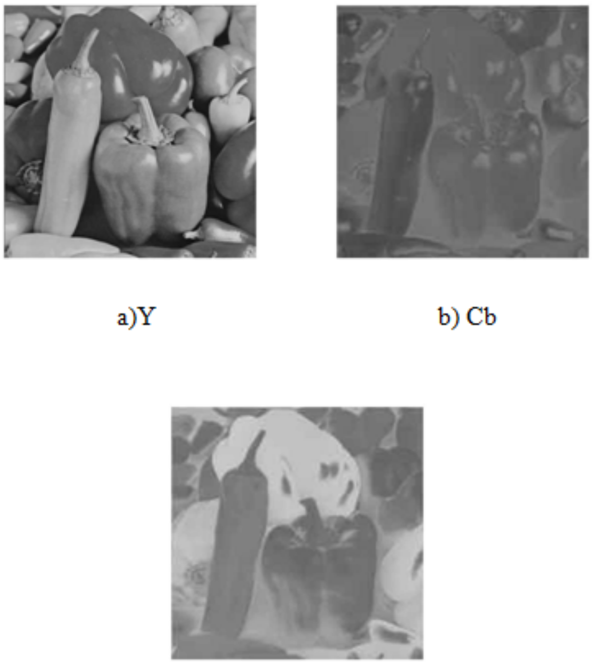
Figure 14. Result after added 128 to component Y, Cb and Cr

After getting the image decoding result, the value of the matrix in each component is then multiplied by the quantization matrix used previously. Purpose of the de quantization is to get back a matrix that contains bits - bits of DCT coefficients. The result can be seen in the Figure 12

After the de quantization of each element of the image will be in inverse DCT, the aim is to restore the source image pixel value that previously has been transformed. The results are shown in the Figure 13.

Inverse DCT matrix has returned the value of constituent image. Furthermore, each element image was added by 128. The result are shown in the Figure 14. After obtained, the matrix making up the image of the final step is to merge elements of Y, Cb and Cr to obtain an image reconstruction close to the image of origin, but the image of Cb and Cr should be made into a matrix of the same size. The results are shown in the Figure 15.

Now each component has the same size, the next step is to merge the three components into YCbCr image. The results can be seen in the Figure 16. The final step is to change the image back to RGB color space. The result is shown in the Figure 17. Thus the decompression process produces an image that looks not much different from the original image, but the size of the image decompression only occupy memory of 30 Kb while the original image has a size of 768 Kb of memory. Another image in the extreme groups can be seen in the Figure 18.



a)Y b) Cb

C)Cr

Figure 15. Result after enlarged Cb and Cr compared with Y

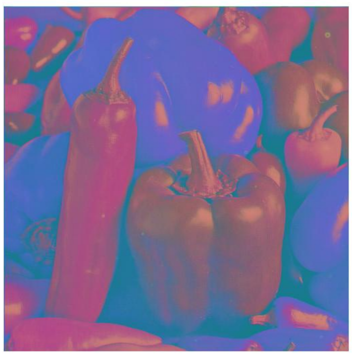


Figure 16. Image Reconstruction in YCbCr Color Space

Figure 17. Image Decompression

Figure 18. Image with extreme color (Source : Picsdesktop.com)

Table 1. Compression Results for Images with Extreme Color

No	Original Image	Initial Memory (Kb)	Result of Compression	Memory Result (kb)	Compression Percentage
1	Figure 1. bmp	768	Figure 1. jpg	31.6	95.88
2	Figure 2. bmp	768	Figure 2. jpg	73.1	90.48
3	Figure 3. bmp	768	Figure 3. jpg	68.0	91.15
4	Figure 4. bmp	768	Figure 4. jpg	47.5	93.82
5	Figure 5. bmp	768	Figure 5. jpg	47.7	93.79
6	Figure 6. bmp	768	Figure 6. jpg	40.6	94.71
7	Figure 7. bmp	768	Figure 7. jpg	46	94.01
8	Figure 8. bmp	768	Figure 8. jpg	46.1	93.99
9	Figure 9. bmp	768	Figure 9. jpg	53.1	93.08
10	Figure 10. bmp	768	Figure 10. jpg	41.4	94.61

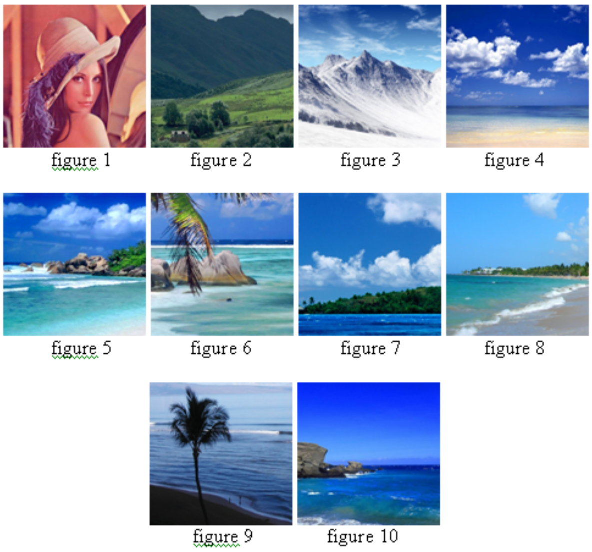


Figure 19. Image with soft color (Source : Picsdesktop.com)

Table 2. compression results for images with soft color

No	Original Image	Initial Memory (Kb)	Result of Compression	Memory Result (kb)	Compression Percentage
1	Figure 1. bmp	768	Figure 1. jpg	29.2	96.20
2	Figure 2. bmp	768	Figure 2. jpg	35.4	95.39
3	Figure 3. bmp	768	Figure 3. jpg	33.0	95.70
4	Figure 4. bmp	768	Figure 4. jpg	27.2	96.46
5	Figure 5. bmp	768	Figure 5. jpg	28.4	96.30
6	Figure 6. bmp	768	Figure 6. jpg	23.1	96.70
7	Figure 7. bmp	768	Figure 7. jpg	18.5	97.59
8	Figure 8. bmp	768	Figure 8. jpg	29.2	96.20
9	Figure 9. bmp	768	Figure 9. jpg	21.0	97.26
10	Figure 10. bmp	768	Figure 10. jpg	29.5	96.16

After compression, the image has a jpg format., the result of compression to all images with extreme colors can be seen in the table. From the Table 1 it can be seen that the percentage of compression for each image with extreme colors achieves almost 96% while the resulting image when viewed with the eyes are not different with the original image.

While for image with a soft color difference can be seen in the Figure 19. The pictures is the image source with a combination of colors in one frame that is almost homogeneous or with soft colors where the color of one another do not have a striking difference. Results of compression for images with a soft color (almost homogeneous) can be seen in the Table 2 .

From the above table it can be seen that the percentage of compression for each image with soft colors achieved 97.59%

CONCLUSION

Based on the result and discussion in the research, it can be concluded that compression using transform dct and quantization have good compression. Image compression obtained using the dct and quantization standard jpg does not damage the image and give good results.

The resulting image when viewed with the eyes doesn't have any difference with the original image but has a much smaller size. And will look a little different if the image compression result is compared with the original image. The result showed the percentage of compression achieved 95.88 percent for extreme image and achieved 97.59% for the soft color (homogeneous). From this comparison, using the method of dct compression according to the results of tests performed, it would give better results when used to compress a color image of a gentle and suitable for images of natural views and sea.

REFERENCES

[1] Robin, Linda. 2001. Menguasai Pembuatan animasi dengan Macromedia Flash. Jakarta: Elek Media Komputindo.

[2] Jain, Deepak K., Gaur,D., Gaur, K., and Jain, Neha. 2014." Image Compression using Discrete Cosine Transform and Adaptive Huffman Coding" . *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*.Vol.3, No.1, pp.90 – 94.

[3] A. K. Jain. 1989. Fundamentals of Digital Image Processing. Englewood Cliffs, NJ : Prentice–Hall.

[4] Ronald A. DeVore, Bjorn Jawerth, and Bradley J. Lucier, Member. 1992. "Image Compression Through Wavelet Transform Coding". *IEEE Trans. on Information Theory*. Vol. 38, No.2, pp. 719-746.

[5] More, Nitesh Kumar and Dubey, Sipi. 2013." JPEG Picture Compression Using Discrete Cosine Transform". *International Journal of Science and Research (IJSR)*.Vol.2, No.1, pp 134 - 138.

[6]. S. Anitha. 2011."Image Compression Using Discrete Cosine Transform & Discrete Wavelet Transform". *International Journal of Scientific & Engineering Research* . Vol.2, No. 8, pp.1 -6.