

Enhancement of Video Data Compression Algorithm Performance by Integrating Discrete Cosine Transform and Quantization Process

Suryadi, Bagyo Budiardjo, and Kalamullah Ramli

Abstract—In this paper will be described integration of Discrete Cosine Transform (DCT) process and quantization in video data compression algorithm. The integration is done in order to improve time and compression ratio. The method is employed by inserting quantization process in the DCT function, as well as for the decompression process. It can be done since the orthonormality property of the prior DCT function is still valid, algebraically. Result obtained in this paper is related with the first aim, that is, a better compression time. It is obtained based on mathematical and algorithm analyses done, that the integration of the two processes can improve to a better time complexity.

Index Terms—DCT, integration of DCT and quantization, quantization, video compression algorithm.

I. INTRODUCTION

Video compression algorithms have been developed and even standardized. Standardization is needed in order to facilitate the storage and exchange of digital video data globally. For multimedia communications, there are two main standards organization which are the International Telecommunication Union - Telecommunication Standardization Sector (ITU-T) and the International Organization for Standardization (ISO).

Recently, a number of ITU-T standards have been developed for many application domains such as H.261, H.263, H.264/AVC [1]-[3]. Similarly, a number of ISO standards have been developed such as MPEG-1, MPEG-2 and MPEG-4 [4]-[5]. These standards define the bitstream of audio-visual data and determine a set of rules to be followed in the development of hardware and software for optimal compression solutions.

Compression techniques that were developed in the majority of those algorithms were using Discrete Cosine Transform (DCT) process, and followed by quantization process sequentially. Since they were done in sequence, then the computation times became longer.

Considering those conditions, they give the idea of how to enhance the performance of video data compression, in the sense of its computation time and compression rate. It can be done by combining the DCT process and quantization, and the quantizer is function-based valued. That study is part of

the main research done that is enhancement of video encryption performance using finite field Z_2^3 in its bit stream basis [6]. The algorithm's improvement is started by compression process which is discussed in this paper, and continued by encryption process using finite field Z_2^3 -based bit stream.

In this paper we describe the process of combining DCT and quantization processes based on algebraic concept. In Section 2 we explained mathematical aspects of the DCT and quantization processes in video compression algorithm. The process of integrating the DCT and quantization processes using algebraic concept, as well as the algorithm analysis are presented in Section 3. This paper ended with Conclusion, stated in Section 4.

II. VIDEO COMPRESSION ALGORITHM

Compression algorithms that are chosen as the standard in video data compression process by many researchs are H.261 and MPEG4 algorithms. The general form of the process in given in Fig. 1.

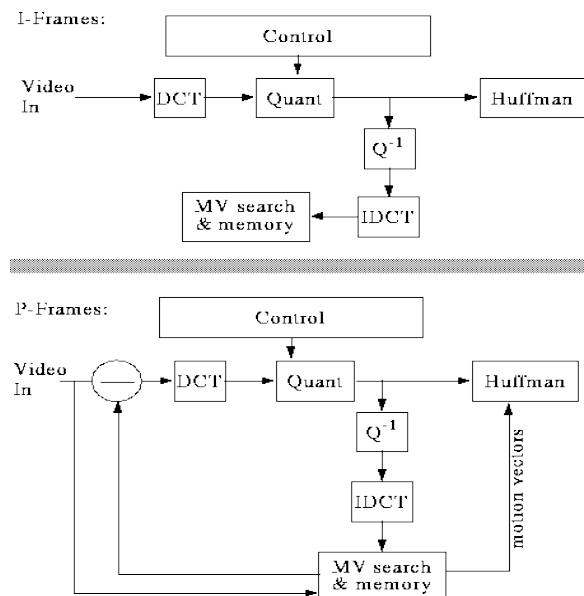


Fig. 1. Compression process diagram [1].

In that video compression process, generally there are data alterations in part I-frame and P-frame using DCT and quantization processes consecutively.

This paper will be focused on the effort of integrating the DCT and quantization processes, which is expected that it can reduce the computation time and data size, based on mathematics analysis. However, prior to that, explanation about the DCT and quantization processes will be explained

Manuscript received August 3, 2012; revised September 23, 2012.

Suryadi is with the Department of Mathematics, and the Department of Electrical Engineering, Universitas Indonesia, Depok, Indonesia, 16424 (e-mail: suryadi81@ui.ac.id, yadi.mt@sci.ui.ac.id).

B. Budiardjo and K. Ramli are with the Department of Electrical Engineering, Universitas Indonesia, Depok, Indonesia, 16424 (e-mail: bbudi@ee.ui.ac.id, k.ramli@ee.ui.ac.id).

mathematically based on the DCT formulation and function-based quantization.

In general, the DCT process in to change the data from spatial data form into frequency data. The input for the DCT process is video data in the form of video image in two dimensional form of size $N \times M$, that is data in spatial domain $f(i,j)$ which represents the intensity of the corresponding pixel in the i -th row and the j -th column. The output of the DCT process is in the form of matrix of size $N \times M$, where the entries are data in frequency domain. Therefore, data with high frequency, as well as low frequency data, are obtained. This fact give result of separation between global information (which is in low frequency) and detail information (which is in high frequency) of every image in the video.

The general form of the DCT is:

$$F(u, v) = \left(\frac{2}{N}\right)^{\frac{1}{2}} \left(\frac{2}{M}\right)^{\frac{1}{2}} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} C(i)C(j) \cos \left[\frac{\pi u}{2N} (2i + 1) \right] \cos \left[\frac{\pi v}{2M} (2j + 1) \right] f(i, j) \quad (1)$$

where:

$$C(i) = \begin{cases} \frac{1}{\sqrt{2}} & \text{untuk } i = 0 \\ 1 & \text{untuk } i \neq 0 \end{cases} \quad (2)$$

In the process, if it is executed in 1-dimensional form (1-D), then the form of equality (1) can be represented in simultaneous equations as the following:

$$F(u, j)_i = \left(\frac{2}{N}\right)^{\frac{1}{2}} \sum_{i=0}^{N-1} C(u) \cos \left[\frac{\pi u}{2N} (2i + 1) \right] f(i, j) \quad (3)$$

$$F(u, v) = \left(\frac{2}{M}\right)^{\frac{1}{2}} \sum_{j=0}^{M-1} C(v) \cos \left[\frac{\pi v}{2M} (2j + 1) \right] F(u, j)_i \quad (4)$$

According to equation (1), it is seen that the DCT process is a multiplication of the image matrix where the entries are $f(x,y)$, with cosines equalities matrix, where in this case, the cosines matrix $[DC]_{uv}$ is of the form: [7],

$$\begin{cases} \frac{1}{\sqrt{N}} & \text{untuk } u = 0, \quad 0 \leq v \leq N-1 \\ \sqrt{\frac{2}{N}} \cos \frac{(2v+1)u\pi}{2N} & \text{untuk } 1 \leq u \leq N-1, \quad 0 \leq v \leq N-1 \end{cases} \quad (5)$$

Therefore, equalities (3) and (4) can be represented as matrices multiplication as the following:

$$[F(u, y)_x] = [DC] \times [f(x, y)] \quad (6)$$

$$[F(u, v)] = [F(u, y)_x] \times [DC]^T \quad (7)$$

As for video compression in image frame part, it is the same as in image data, where the size of standard matrix is 8×8 pixel size ($N=8$). Therefore, matrix DC can be evaluated according to equation (5) and matrix DC is obtained as the following:

$$\text{For every } u = 0, \text{ we will obtain } DC(0,0) = DC(0,1) = DC(0,2) = DC(0,3) = DC(0,4) = DC(0,5) = DC(0,6) = DC(0,7) = \frac{1}{\sqrt{8}} = 0,3536$$

Otherwise, the formula $\sqrt{\frac{2}{N}} \cos \frac{(2v+1)u\pi}{2N}$ is used, for example: for $u = 1$ and $v = 0$, we will obtain $DC(1,0) =$

$$\sqrt{\frac{2}{8}} \cos \frac{(0+1)\pi}{2.8} = \frac{1}{2} \cos \frac{\pi}{16} = 0.4904$$

Using similar procedure, the entry of $DC(1,1)$ up to the last entry, $DC(7,7)$ will be obtained. The complete DC matrix is given below:

$$[DC]_{8 \times 8} = \begin{bmatrix} 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 \\ 0.4904 & 0.4157 & 0.2778 & 0.0975 & -0.0975 & -0.2778 & -0.4157 & -0.4904 \\ 0.4619 & 0.1913 & -0.1913 & -0.4619 & -0.4619 & -0.1913 & 0.1913 & 0.4619 \\ 0.4157 & -0.0975 & -0.4903 & -0.2778 & 0.2778 & 0.4903 & 0.0975 & -0.4157 \\ 0.3536 & -0.3536 & -0.3536 & 0.3536 & 0.3536 & -0.3536 & -0.3536 & 0.3536 \\ 0.2778 & -0.4903 & 0.0975 & 0.4157 & -0.4157 & -0.0975 & 0.4903 & -0.2778 \\ 0.1913 & -0.4619 & 0.4619 & -0.1913 & -0.1913 & 0.4619 & -0.4619 & 0.1913 \\ 0.0975 & -0.2778 & 0.4157 & -0.4903 & 0.4903 & -0.4157 & 0.2778 & -0.0975 \end{bmatrix}$$

Based linear algebraic concept, by Howard Anton [8], orthonormal vectors and orthogonal matrix are defined as the following:

Definition 1. Two vectors u and v in an inner product space are called *orthogonal* if $\langle u, v \rangle = 0$.

Definition 2. A set of vectors in an inner product space is called an *orthogonal set* if all pairs of distinct vectors in the set are orthogonal. An orthogonal set in which each vector has norm 1 is called *orthonormal*.

Now, consider the 1st and the 2nd columns of $[DC]$, dot product of the two columns is 0.000. For every pair of column in $[DC]$, it will be obtained that the result of all dot product is 0.000. In addition, norm of k1 is 1.

Using similar procedure, it can be shown that the norm of each column of $[DC] = 1$. According to Definition 1 and 2, every column of DC is orthogonal to each other column, so that $[DC]$ is an orthogonal matrix and every column in $[DC]$ forms a set of orthonormal vectors. This fact implies that $[DC]$ is an invertible matrix, and the inverse is its transpose. Therefore, matrix $[DC]_{8 \times 8}$ can be used in compression process, while for the decompression process, the inverse of $[DC]_{8 \times 8}$ will be used, that is:

$$[DC]^{-1} = \begin{bmatrix} 0.3536 & 0.4904 & 0.4619 & 0.4157 & 0.3536 & 0.2778 & 0.1913 & 0.0975 \\ 0.3536 & 0.4157 & 0.1913 & -0.0975 & -0.3536 & -0.4904 & -0.4619 & -0.2778 \\ 0.3536 & 0.2778 & -0.1913 & -0.4904 & -0.3536 & 0.0975 & 0.4619 & 0.4157 \\ 0.3536 & 0.0975 & -0.4619 & -0.2778 & 0.3536 & 0.4157 & -0.1913 & -0.4904 \\ 0.3536 & -0.0975 & -0.4619 & 0.2778 & 0.3536 & -0.4157 & -0.1913 & 0.4904 \\ 0.3536 & -0.2778 & -0.1913 & 0.4904 & -0.3536 & -0.0975 & 0.4619 & -0.4157 \\ 0.3536 & -0.4157 & 0.1913 & 0.0975 & -0.3536 & 0.4904 & -0.4619 & 0.2778 \\ 0.3536 & -0.4904 & 0.4619 & -0.4157 & 0.3536 & -0.2778 & 0.1913 & -0.0975 \end{bmatrix}$$

According to linear algebraic concept that if matrix A has an inverse, say A^{-1} , then $AA^{-1} = A^{-1}A = I$ (identity matrix). Moreover, matrix multiplication has an associative property. Therefore, for matrices DC and DC^{-1} , the following property holds: $[DC]_{8 \times 8} \times [DC]_{8 \times 8}^{-1} = [DC]_{8 \times 8}^{-1} \times [DC]_{8 \times 8} = [I]$. This property implies that the original image can be re-obtained, in accordance with the form matrices multiplication $[DC]_{8 \times 8}^{-1} \times ([DC]_{8 \times 8} \times [f(x, y)]) = ([DC]_{8 \times 8}^{-1} \times ([DC]_{8 \times 8})) \times [f(x, y)] = [I] \times [f(x, y)] = [f(x, y)]$

In order to prove the above mentioned fact, if $[DC]$ is multiplied with $[DC]^{-1}$, then the following matrix will be obtained:

$$[DC]_{8 \times 8} * [DC]_{8 \times 8}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

On the other hand, if matrix $[DC]_{8 \times 8}$ is multiplied with the

image matrix:

$$[f(x,y)] = \begin{bmatrix} 164 & 63 & 75 & 95 & 157 & 99 & 91 & 51 \\ 120 & 135 & 55 & 75 & 116 & 67 & 84 & 49 \\ 99 & 132 & 60 & 54 & 100 & 75 & 72 & 58 \\ 64 & 150 & 113 & 50 & 81 & 138 & 70 & 53 \\ 110 & 130 & 162 & 60 & 76 & 109 & 86 & 65 \\ 97 & 82 & 179 & 81 & 74 & 113 & 64 & 56 \\ 61 & 70 & 149 & 159 & 62 & 138 & 67 & 68 \\ 54 & 63 & 115 & 187 & 95 & 141 & 51 & 72 \end{bmatrix}$$

then it is obtained that:

$$[DC]_{8x8} * [f(x,y)] = \begin{bmatrix} 271.8826 & 291.6815 & 321.0265 & 269.0541 & 269.0541 & 311.1270 & 206.8287 & 166.8772 \\ 74.5400 & 42.8629 & -96.5309 & -88.5135 & 60.5642 & -57.8407 & 27.3448 & -18.8122 \\ 17.4552 & -72.8608 & -45.9618 & 98.3965 & 44.6496 & 0.0192 & -3.5970 & 2.8837 \\ 51.7730 & -36.4158 & 64.5080 & -14.0354 & 6.3690 & 0.0440 & 15.4925 & -4.5244 \\ 5.3033 & -4.5962 & 7.7782 & 8.1317 & 20.1525 & 33.2340 & 3.8891 & 3.5355 \\ -17.3055 & -18.6836 & 3.0066 & 8.8457 & -4.6437 & 31.5005 & -3.0967 & -1.3098 \\ 15.3481 & -25.3092 & -0.0962 & -12.8213 & 16.3297 & -9.1924 & -9.6079 & -0.4291 \\ 17.7302 & -7.0771 & -3.2332 & 8.0389 & -0.5955 & -14.3935 & 10.3516 & 9.9457 \end{bmatrix}$$

Moreover, if the inverse matrix $[DC]_{8x8}^{-1}$ is multiplied with the above resulting matrix, then it will be obtained that:

$$[DC]_{8x8}^{-1} * ([DC]_{8x8} * [f(x,y)]) = \begin{bmatrix} 164 & 63 & 75 & 95 & 157 & 99 & 91 & 51 \\ 120 & 135 & 55 & 75 & 116 & 67 & 84 & 49 \\ 99 & 132 & 60 & 54 & 100 & 75 & 72 & 58 \\ 64 & 150 & 113 & 50 & 81 & 138 & 70 & 53 \\ 110 & 130 & 162 & 60 & 76 & 109 & 86 & 65 \\ 97 & 82 & 179 & 81 & 74 & 113 & 64 & 56 \\ 61 & 70 & 149 & 159 & 62 & 138 & 67 & 68 \\ 54 & 63 & 115 & 187 & 95 & 141 & 51 & 72 \end{bmatrix} = [f(x,y)]$$

The above result shows that these two matrices $[DC]$ and $[DC]^{-1}$ will be used in compression and decompression algorithm.

The resulting matrix from the DCT process inherent two important information: global information from 8x8 pixel image block which is located in the 1st entries of the matrix, known as DC component; while the other 63 entries represented detailed data, known as AC component. In general, values of entries of AC are getting smaller and approximately zero as the columns toward the right and rows toward lower positions. This explains that these values will not have much effect visually if those small values are set as zeroes. Therefore, it can minimize bit number, and increase the compression ratio. Since low frequency and high frequency data are dispersed but concentrated into two main parts, then the next process of compression becomes easier to be done, bit number in detailed information of image's luminance and chrominance will be minimized. This can be done since human's eyes visual system is very sensitive towards the change in global information than that in detailed information. The process is known as quantization.

Numerically, quantization process is the process of changing detail information as zero, by conducting division operation among all entries of matrix DCT. In general, the formulation of quantization process is done using floor function, as the following:

$$P'[u,v] = \left\lfloor \frac{P[u,v]}{q[u,v]} \right\rfloor \quad (8)$$

where:

$P[u,v]$: pixel value of matrix DCT in position (u,v)

$q[u,v]$: entry of quantization matrix in position (u,v)

$P'[u,v]$: pixel score that is quantized in position (u,v) .

Therefore, division operations are needed, where the number of operation is as much as the number of image pixel in the compressed video. For example, if the video size is 30 megapixels, then 30 million divisions operation are required

in quantization process. This is a really computational time consuming process.

Time complexity for the DCT process with block size of 8x8 needs $2x(8x8x8) = 1024$ or $2xN^3$ divisions operation, and $2x(8x8x7) = 896$ or $2xN^2x(N-1)$ additions operation. While for quantization process, it requires $8x8 = 64$ or N^2 divisions operation. Therefore, for image of size of KxL (K is the length and L is the height of the video image), then the DCT and quantization processes for this image requires $KxLx(2N+1)$ multiplications/divisions operation, and $KxLx2(N-1)$ additions/subtractions operation. The sum of those operations is very big, relatively, so that it boosts out the question, as well as the problem in this research, that is it possible to reduce the number of multiplications and additions operation greatly, without affecting the quality of the compressed image?

Considering the above fact, an effort to enhance the performance, in term of its computation time efficiency, by combining the DCT and quantization processes as matrix QDCT will be done. Efficiency effort using matrix QDCT has been done for image data in JPEG format [9].

III. INTEGRATION OF DCT PROCESS AND QUANTIZATION

As has been described in the previous section that the proposed model in this research is combination of the DCT and quantization processes in video data compression and decompression algorithm. The steps done are:

A. Construct a Quantized-DCT Formula

A quantified-DCT formula (DCTQ) can be done simultaneously with the DCT and quantization processes, as shown in Fig. 2(b).

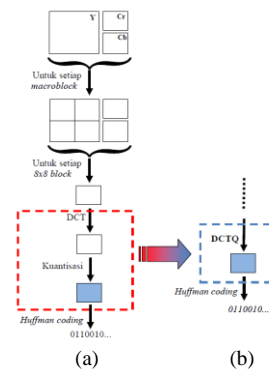


Fig. 2. Compression process (a). standard, (b). integration of dct and quantization

In order to combine the DCT and quantization processes as one quantized-DCT process, then a constructed function (which represents quantization process) is inserted in the DCT formula. This is done since a quantization process is a process of division of DCT with a constant, so that it will not affect the decomposition of function or DCT formula.

As for the DCTQ (DCT-quantized) transformation formulation in 1-D implementation is:

$$DCTQ(u,y)_x = \sqrt{\frac{2}{N}} C(u) \sum_{x=0}^{N-1} f(x,y) \cos\left(\frac{(2x+1)u\pi}{2N}\right) \quad (9)$$

$$F(u,v) = DCTQ(u,v) = \sqrt{\frac{2}{N}} C(v) Q(u,v) \sum_{y=0}^{N-1} DCTQ(u,y) \cos\left(\frac{(2y+1)v\pi}{2N}\right) \quad (10)$$

where $Q(u,v)$ is a quantization function.

The matrices multiplication of DCTQ transformation formulations are:

$$[DCTQ(u,y)_x] = [DCQ] \times [f(x,y)] \quad (11)$$

$$[F(u,v)] = [DCTQ(u,v)] = [F(u,y)_x] \times [DCQ]^T \quad (12)$$

$[DCQ]$ is the result of quantization of matrix $[DC]$, where each entry of $[DC]$ is divided by the entry on the same position in the same quantized matrix, dan this process is done in the beginning of the DCT process.

B. Construct a Quantized-DCT Invers Formula

A DCT-quantized inverse (IDCTQ) formulation can perform the dequantization and DCT inverse processes simultaneously. Similar with the procedure to find a quantized-DCT, to combine dequantized process and DCT inverse as one process, a quantized-DCT inverse process, then an inverse function is inserted, which is used in quantized-DCT process in inverse DCT formula. The formulations of IDCTQ (a quantized-DCT inverse) transformation which are implemented in 1-D are:

$$F^{-1}(u,y)_x = \sqrt{\frac{2}{N}} C(u) Q^{-1}(u,v) \sum_{v=0}^{N-1} F(u,v) \cos\left(\frac{(2x+1)v\pi}{2N}\right) \quad (13)$$

$$f(x,y) = \sqrt{\frac{2}{N}} C(v) \sum_{u=0}^{N-1} F^{-1}(u,y)_x \cos\left(\frac{(2y+1)u\pi}{2N}\right) \quad (14)$$

where $Q^{-1}(u,v)$ is a constant.

As for the multiplication of matrix from IDCTQ (a quantized-DCT inverse) process is:

$$[DCTQ^{-1}(u,y)_x] = [[DCQ]^T]^{-1} \times [F(u,v)] \quad (15)$$

$$[f(x,y)] = [DCQ]^{-1} \times [DCTQ^{-1}(u,y)] \quad (16)$$

As for decompression process, $[DCQ]$ and $[DCQ]^T$ each has to be invertible. Since $[DC]$ is an orthogonal matrix, then it is invertible. Considering its properties, when an inverse matrix is multiplied by a constant, then that matrix is still invertible. $[DCQ]$ is a result of multiplication of $[DC]$ with a constant, so that $[DCQ]$ has an inverse. $[QDC]^T$ is the transpose of $[QDC]$, then $[QDC]^T$ has an inverse, and the following hold: $[DCQ] \times [DCQ]^{-1} = [I]$ and $[DCQ]^T \times [DCQ]^T^{-1} = [I]$

In this case, the difference of compression process using a standard DCT and a quantized-DCT ($DCTQ$) is in the quantization process. For a standard DCT, the DCT and quantization processes are done in every block, while for a quantized DCT, the DCT is done only once when calculating $[DCQ]$ at the beginning of the process. The formula will do the DCT transformation process and quantization process simultaneously. Therefore, DCTQ process will accelerate compression process since it can reduce number of division process.

Number of operation done in a quantized-DCT algorithm is:

- 1) for matrix DCQ initialization (quantization of matrix DC), it needs $8 \times 8 = 64$ or N^2 division operation (only at the beginning of the process).
- 2) for DCTQ process, it needs $2 \times (8 \times 8 \times 8) = 1024$ or $2 \times N^3$ multiplications operation and $2 \times (8 \times 8 \times 7) = 896$ or $2 \times N^2 \times (N-1)$ additions operation.

Therefore, when a video image of size $K \times L$ pixels, then a quantized DCT process for this image requires $K \times L \times 2N + N^2$

multiplications/divisions operation and $K \times L \times 2(N-1)$ additions/subtractions operation. The same condition applies for decompression process. So, based on algorithm analysis by calculating its time complexity, the number of operations performed by a quantized DCT algorithm is less than that of a standard DCT algorithm. Therefore, time execution needed for compression and decompression using a quantized-DCT algorithm is faster than a standard DCT algorithm.

IV. CONCLUSION

According to linear algebraic concept, DCT and quantization processes can be integrated as one process, and a quantization function can be inserted in DCT function at the same time. This condition also applies for decompression process. The consequence of integrating those two processes, by performing algorithm analysis (its time complexity), it can be calculated that number of operations done in a quantized-DCT algorithm is less than that in a standard DCT algorithm. It implies that execution time needed for compression and decompression processes using a quantized DCT algorithm is faster than a standard DCT algorithm.

REFERENCES

- [1] Dave Marshal. H.261 Compression. [Online]. Available: <http://www.cs.cf.ac.uk/Dave/Multimedia/node246.html>
- [2] ITU-T, "Video Coding for Low Bit Rate Communication," ITU-T Recommendation H.263, 1998.
- [3] G. J. Sullivan, P. Topiwala, and A. Luthra, "The H.264/AVC advanced video coding standard: Overview and introduction to the fidelity range extensions," *SPIE Conference on Applications of Digital Image Processing XXVII*, 2004.
- [4] T. Sikora, "MPEG-1 and MPEG-2 digital video coding standards," *Digital Consumer Electronics Handbook, McGraw-Hill Book Company*.
- [5] MPEG 1 and MPEG 2 Digital Video Coding Standards. [Online]. Available: <http://www.di.univr.it/documenti/OccorrenzaIn/madid/matdid279434.pdf>
- [6] "MPEG digital video-coding standards," *IEEE Signal Processing Magazine*, Sept. 1997, pp. 82-100.
- [7] Suryadi, B. Budiardjo, and K. Ramli, "Enhancement of video encryption algorithm performance using finite field Z_2^3 -based chaotic cipher," *Proc. 3th-The International Symposium on Chaos Revolution in Science, Technology and Society*, 2011.
- [8] K. Cabeen and P. Gent, "Image compression and the discrete cosine transform," *College of the Redwoods*.
- [9] Anton and Howard, "Elementary linear algebra," *John Wiley and Sons, Inc., Canada*, 2005. pp. 307-322.
- [10] E. Sukirman, Ernastuti, and S. dan Madenda, "Peningkatan kinerja algoritma kompresi dan dekompresi JPEG melalui penggabungan proses DCT dan kuantisasi," *Proc. KOMMIT 2010*.



Suryadi was born on October 20, 1965 in Bima. He is an undergraduate in mathematics, master in informatics, and PhD student in electrical engineering. Lecture in Department of Mathematics, Faculty of Mathematics and Natural Sciences Universitas Indonesia, (1990, -). research fields: cryptography, computational mathematics information & security system, Publishers of a book : *Dasar-dasar Ilmu Komputer*, Depok, Departemen Matematika FMIPA UI, 2011 ISBN:

978-602-19307-0-0. Paper : Suryadi, B. Budiardjo, and K. Ramli. (2011). Enhancement of Video Encryption Algorithm Performance using Finite Field Z_2^3 -based Chaotic Cipher and by Integrating Discrete Cosine Transform and Quantization Process. *Proceedings of the 10th WSEAS International Conference on Information Security and Privacy (ISP '11)* : pp. 168 – 173. ISBN: 978-1-61804-049-7 Drs., M.T., Member of IndoMS (The Indonesian Mathematical Society)