1-1-2006

Recovering DC coefficients in block-based DCT

Takeyuki Uehara

University of Wollongong, takeyuki@uow.edu.au

Reihaneh Safavi-Naini

University of Wollongong, rei@uow.edu.au

Philip Ogunbona

University of Wollongong, philipo@uow.edu.au

Follow this and additional works at: <https://ro.uow.edu.au/infopapers>

Part of the [Physical Sciences and Mathematics Commons](#)

Recommended Citation

Uehara, Takeyuki; Safavi-Naini, Reihaneh; and Ogunbona, Philip: Recovering DC coefficients in block-based DCT 2006, 3592-3596.

<https://ro.uow.edu.au/infopapers/2121>

Recovering DC coefficients in block-based DCT

Abstract

It is a common approach for JPEG and MPEG encryption systems to provide higher protection for dc coefficients and less protection for ac coefficients. Some authors have employed a cryptographic encryption algorithm for the dc coefficients and left the ac coefficients to techniques based on random permutation lists which are known to be weak against known-plaintext and chosen-ciphertext attacks. In this paper we show that in block-based DCT, it is possible to recover dc coefficients from ac coefficients with reasonable image quality and show the insecurity of image encryption methods which rely on the encryption of dc values using a cryptoalgorithm. The method proposed in this paper combines dc recovery from ac coefficients and the fact that ac coefficients can be recovered using a chosen ciphertext attack. We demonstrate that a method proposed by Tang to encrypt and decrypt MPEG video can be completely broken.

Keywords

dc, block, recovering, dct, coefficients

Disciplines

Physical Sciences and Mathematics

Publication Details

Uehara, T., Safavi-Naini, R. & Ogunbona, P. (2006). Recovering DC Coefficients in Block-Based DCT. IEEE Transactions on Image Processing, 15 (11), 3592-3596.

Correspondence

Recovering DC Coefficients in Block-Based DCT

Takeyuki Uehara, Reihaneh Safavi-Naini, *Senior Member, IEEE*,
and Philip Ogunbona, *Senior Member, IEEE*

Abstract—It is a common approach for JPEG and MPEG encryption systems to provide higher protection for dc coefficients and less protection for ac coefficients. Some authors have employed a cryptographic encryption algorithm for the dc coefficients and left the ac coefficients to techniques based on random permutation lists which are known to be weak against known-plaintext and chosen-ciphertext attacks. In this paper we show that in block-based DCT, it is possible to recover dc coefficients from ac coefficients with reasonable image quality and show the insecurity of image encryption methods which rely on the encryption of dc values using a cryptotransformation. The method proposed in this paper combines dc recovery from ac coefficients and the fact that ac coefficients can be recovered using a chosen ciphertext attack. We demonstrate that a method proposed by Tang to encrypt and decrypt MPEG video can be completely broken.

Index Terms—Block DCT, image coding, image encryption, image reconstruction, multimedia data security.

I. INTRODUCTION

The discrete cosine transform (DCT) [1] is a signal-analysis tool that can be used to decompose an image signal into its frequency components. The energy compaction efficiency of the DCT is known to be near-optimal for autoregressive order-1 (AR-1) signals and is widely used in the decomposition stage of natural image compression systems [2]. For example, it is the transform of choice in JPEG [3], MPEG2 [4], and H.261 coding standards. For efficient computation of transform coefficients, the image is partitioned into blocks of subimages and the transform is applied to each block independently. The transform coefficients can be classified into two groups namely, dc and ac coefficients. The dc coefficient is the mean value of the image block and carries most of the energy in the image block. The ac coefficients carry energy depending on the amount of detail in the image block. However, most of the energy is compacted in the dc coefficient and a few ac coefficients. Image compression systems exploit the energy compaction property of DCT and, because of the small amount of energy in the higher frequency ac coefficients, use quantization to produce a more compact representation of the image.

Adding security to compression system is an attractive proposition that can reduce the overall cost of compression and encryption and various image/video encryption systems have been proposed [5]–[11]. Proposals by Tang [12] and Shin *et al.* [13] that incorporate encryption in compression systems apply a secret permutation only known to the sender and the receiver, to DCT coefficients. However, in each block dc coefficient carries most of the signal energy and is much larger than the other coefficients and so it is easily distinguishable in the permuted form. Knowing the dc coefficients of the blocks, a low-resolution version of the image can be reconstructed and so direct permutation of the

coefficients cannot make the image incomprehensible. To avoid this, a cryptographic encryption algorithm is used to encrypt the dc coefficients of blocks while permuting the ac coefficients. The method is claimed to produce images that are *incomprehensible*, and so the scrambled images do not leak any information to outsiders. However, there are attacks [14], [15] which find the correct order of ac coefficients from the permuted ones. If a method to recover the dc coefficients from the ac coefficients can be devised, then by combining the ac recovery attacks and the dc recovery method, these encryption systems using a secret permutation can be broken.

We motivate this paper by posing the question: Is it possible to recover the dc coefficients of image blocks when only the ac coefficients are known? Furthermore, what is the image quality performance sacrificed by using dc coefficient estimates in image reconstruction? We answer these questions in the following sections. In Section II, we develop some of the known properties of DCT useful for the algorithmic development along with the dc recovery algorithm, and Section III reports on the results of computer simulations. In Section IV, an attack on MPEG encryption is introduced and Section V is the conclusion.

II. RECOVERING DC COEFFICIENT IN BLOCK-BASED DCT

A. Block-Based DCT

Let the pixel values $x_{i,j}$ in a given $N \times N$ image block be represented as the matrix $[X]$. Further, let the maximum possible value of the pixel values in the image be $x_{\max}$ and the smallest quantization increment of a pixel value be $x_{\min}$. The DCT of the data is given as the matrix $[C]_{\text{DCT}}$, $[C]_{\text{DCT}} = [A][X][A]^t$ with entries $c_{i,j}$ and $[A]$ is the matrix of DCT basis vectors. The inverse transform recovers the image block $[X']$ as $[X']_{\text{IDCT}} = [A]^t[C][A]$. The pixel values can be written as a decomposition into a dc and an ac component. Thus

$$[X] = [X]_{\text{dc}} + [X]_{\text{ac}} \quad (1)$$

and $[X]_{\text{dc}}$ is constant over all $i, j \in N$ with its components, given by

$$(x_{\text{dc}})_{i,j} = \frac{1}{N^2} c_{1,1} \quad (2)$$

where $c_{1,1}$ is the dc coefficient of the $N \times N$ image block. Note that in the case of JPEG where $N = 8$, the multiplying factor is $1/64$. The range of the dc coefficient value has been shown [16] to be $\Lambda_{\text{dc}} = (x_{\max}/x_{\min}) \cdot N^2$. The ac coefficients can have both positive and negative values with range given as $\Lambda_{\text{ac}}(i, j) = (x_{\max}/x_{\min}) \cdot (\sum_{n=1}^N |c_{i,n}| \sum_{n=1}^N |c_{j,n}| / |c_{i,n(\min)}| |c_{j,n(\min)}|)$. In essence, the range of the ac coefficient value varies from coefficient to coefficient.

B. Relationship Between DC Coefficients of Neighboring Blocks

The dc coefficients of DCT transform of image blocks represent the mean value of the image blocks. These dc values will reconstruct a decimated (and low-pass filtered) version of the original image. For a natural image, the pixel level correlation structure is carried over to the low-pass filtered version. In the smooth areas of the image, the prediction error of a first-order predictor is small and large prediction errors are only observed along the edges. In general the difference signal at the pixel level and the dc coefficients, has been modeled as a zero-mean

Manuscript received September 14, 2004; revised May 18, 2005. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jianying Hu.

The authors are with School of Information Technology and Computer Science, University of Wollongong, Wollongong, NSW 2522, Australia (e-mail: takeyuki@uow.edu.au; rei@uow.edu.au; philipo@uow.edu.au).

Digital Object Identifier 10.1109/TIP.2006.881939

TABLE I
MEAN AND STANDARD DEVIATION OF $x_{i,j} - q_{i,j} \forall i, j$
AND THE RANGE OF PIXEL VALUES

Image	Mean	Std. dev	Pixel range
airfield256x256.pgm	0.04	33.9	0 - 255
mandrill.pgm	-0.18	34.9	0 - 255
lena.pgm	0.01	11.5	24 - 245
peppers.pgm	-0.35	19.5	0 - 225

Fig. 1. Distribution of $x_{i,j} - q_{i,j} \forall i, j$ in “airfield256 × 256” (left top), “mandrill” (right top), “Lena” (left bottom), and “peppers” (right bottom).

Laplacian distributed variate. The distribution is generally narrow with a small value of variance.

Let $x_{i,j}$ be a pixel at i, j and $Q_{i,j} = \{x_{i+1,j-1}, x_{i+1,j}, x_{i+1,j+1}, x_{i,j+1}, x_{i,j-1}, x_{i-1,j-1}, x_{i-1,j}, x_{i-1,j+1}\}$ be an 8-connected neighbour. Then we call a pair of pixels $(x_{i,j}, q_{i,j})$ as *neighboring pixels* where $q_{i,j} \in Q_{i,j}$. For a natural image, we expect that the distribution of $x_{i,j} - q_{i,j}$ is narrow with a small variance. In Table I, we show the statistics for some sample images:

- the means and standard deviations of $x_{i,j} - q_{i,j} \forall i, j$;
- the ranges of pixel values of the images.

Fig. 1 shows the distribution of $x_{i,j} - q_{i,j} \forall i, j$ for the same sample images.

As shown in (1), the pixel values can be decomposed into dc and ac parts. The dc part is constant while the ac part is the mean-removed pixel value. A large range for the ac value constrains the dc part to a small value since the total range cannot exceed that of the pixel value. We summarize these considerations into two properties.

Property 1: The difference between two neighboring pixels is a Laplacian variate with zero mean and small variance.

Property 2: The range of $[X]_{ac}$, the mean-removed pixel values, constrains the value of $[X]_{dc}$. In particular, a large range for $[X]_{ac}$ implies small values for $[X]_{dc}$. The converse is also true.

The two blocks can be horizontally or vertically adjacent. Let $T^{(k)}$ and $T^{(k+1)}$ denote two adjacent 8×8 pixel blocks and $(p^{(k)}, p^{(k+1)})$ denote a pair of neighboring pixels, where $p^{(k)}$ is in $T^{(k)}$ and $p^{(k+1)}$ is in $T^{(k+1)}$. We note that in subsequent development of our work, the reference to neighboring pixels alludes to pixels at the boundary of two adjacent (vertically and horizontally) blocks.

Then, given the distribution of the difference of two adjacent pixels, the two pixels can be related as

$$p^{(k)} = p^{(k+1)} + \varepsilon \quad (3)$$

where ε has zero mean with small variance. In the following, we will show that when (3) holds, the dc signal value of the two blocks can be related through the values of the ac signals in the two blocks. A related work reported by Jiang and Feng in [17] considered the spatial rela-

tionship of DCT coefficients between a block and its subblocks. They derived algorithm to compose the DCT of a block of pixels from those of its subblocks in the DCT domain. The problem being considered in this paper differs from that in [17] because here, we do not have the dc coefficients and as such the complete DCT coefficients of any part of the image is not available. In the sequel, we develop our method of recovering the dc coefficient by exploiting the inherent properties between adjacent ac coefficients of neighboring blocks in the DCT domain.

Assume the DCT coefficients, $c_{i,j}^{(k)}, i, j \in [1, 2, \dots, 8]$ and $c_{i,j}^{(k+1)}, i, j \in [1, 2, \dots, 8]$ are known, except for the dc coefficient, $c_{1,1}^{(k+1)}$ of block $k+1$. That is, $x_{ac}^{(k)}, x_{dc}^{(k)}$ and $x_{ac}^{(k+1)}$ are known but $x_{dc}^{(k+1)}$ is unknown.

We inverse-transform the ac coefficients of the two blocks to obtain the matrix of dc-free pixels values, $\Gamma^{(k)}$ and $\Gamma^{(k+1)}$ corresponding, respectively, to $x_{ac}^{(k)}$ and $x_{ac}^{(k+1)}$.

Let $(\rho^{(k)}, \rho^{(k+1)})$ denote the pair of neighboring dc-free pixel values corresponding to the pair $(p^{(k)}, p^{(k+1)})$.

From (1) and (2), we have

$$p^{(k)} = \rho^{(k)} + \frac{1}{N^2} c_{1,1}^{(k)}$$

$$p^{(k+1)} = \rho^{(k+1)} + \frac{1}{N^2} c_{1,1}^{(k+1)}.$$

From (3)

$$\rho^{(k)} + \frac{1}{N^2} c_{1,1}^{(k)} \approx \rho^{(k+1)} + \frac{1}{N^2} c_{1,1}^{(k+1)}$$

$$c_{1,1}^{(k+1)} \approx \left(\rho^{(k)} - \rho^{(k+1)} \right) N^2 + c_{1,1}^{(k)}. \quad (4)$$

Hence, since $\rho^{(k)}, \rho^{(k+1)}$, and $c_{1,1}^{(k)}$ are known, the value of $c_{1,1}^{(k+1)}$ can be estimated.

C. Estimating the Dc Coefficient of a Block

If the pixel values of an adjacent block are unknown but their dc-free values are known, then (4) can be used to estimate the dc value of the adjacent block.

Consider dc-free values of pixels in two adjacent blocks. We use (4) on pairs of horizontally neighboring pixels to obtain estimates of $c_{1,1}^{(j+1)}$ from pairs $(\rho_i^{(j)}, \rho_i^{(j+1)})$, $i \in [1, 2, \dots, 8]$, and then obtain a final estimate as the average of all such estimates.

Suppose the j th block is the reference block and let $\Delta_{j+1} = c_{1,1}^{(j+1)} - c_{1,1}^{(j)}$ denote the estimated adjustment for pixels in the $(j+1)$ th block. Then, we use

$$\Delta_{j+1} = \frac{N^2 \sum_{i=1}^8 (\rho_i^{(j)} - \rho_i^{(j+1)})}{8} \quad (5)$$

as the final estimate of the difference and the corrected pixel values $\rho^{(j+1)}$ are calculated as $\rho^{(j+1)} = \rho^{(j)} + 1/N^2 \Delta_{j+1}$.

The noise induced by this estimation can be reduced by taking an average over a number of pixels. This method will perform well only if horizontally (vertically) neighboring pixels in two adjacent columns (rows) have close values. For regions with high variation in horizontal and vertical directions but smooth in the diagonal direction, the algorithm will produce poor estimates. In the following, we modify the algorithm to find the “smoothest” direction among the horizontal, vertical, and two diagonal directions, and use it to find an estimate of the dc value.

The basic idea is to consider three sets of pixel pairs in two adjacent columns (rows) that correspond to horizontal (vertical) and two diagonal directions, and use the mean-square error to choose the smoothest direction.

Fig. 2. Three patterns at the border in the case of horizontally neighboring two blocks.

Let $T^{(k+1)}$ be the block adjacent to $T^{(k)}$ in the horizontal (vertical) direction. The three sets of pixels are: 1) $(\rho_i^{(k)}, \rho_i^{(k+1)}), i \in [1, 2, \dots, 8]$ (pattern 1 in Fig. 2); 2) $(\rho_{i+1}^{(k)}, \rho_i^{(k+1)}), i \in [1, 2, \dots, 7]$ (pattern 2 in Fig. 2); and 3) $(\rho_i^{(k)}, \rho_{i+1}^{(k+1)}), i \in [1, 2, \dots, 7]$ (pattern 3 in Fig. 2). Then the smoothest among the three directions is chosen to estimate the dc value.

Consider three vectors consisting of pixels in $\Gamma^{(k)}$ and three vectors of pixels for $\Gamma^{(k+1)}$ as

$$\begin{aligned}\beta_1^{(k)} &= (\rho_1^{(k)}, \rho_2^{(k)}, \dots, \rho_8^{(k)}) \\ \beta_2^{(k)} &= (\rho_2^{(k)}, \rho_3^{(k)}, \dots, \rho_8^{(k)}) \\ \beta_3^{(k)} &= (\rho_1^{(k)}, \rho_2^{(k)}, \dots, \rho_7^{(k)}) \\ \beta_1^{(k+1)} &= (\rho_1^{(k+1)}, \rho_2^{(k+1)}, \dots, \rho_8^{(k+1)}) \\ \beta_2^{(k+1)} &= (\rho_1^{(k+1)}, \rho_2^{(k+1)}, \dots, \rho_7^{(k+1)}) \\ \beta_3^{(k+1)} &= (\rho_2^{(k+1)}, \rho_3^{(k+1)}, \dots, \rho_8^{(k+1)}).\end{aligned}$$

The above three sets of pairs are $\beta_v^{(k)}$ and $\beta_v^{(k+1)}$, where $v = 1, 2, 3$. Then the method to choose the smoothest direction is as follows.

- i) Calculate the means $M_v^{(k)}$ and $M_v^{(k+1)}$ of $\beta_v^{(k)}$ and $\beta_v^{(k+1)}$ as follows:

$$\begin{aligned}M_1^{(k)} &= \frac{\sum_{n=1}^8 \rho_n^{(k)}}{8}, & M_2^{(k)} &= \frac{\sum_{n=2}^8 \rho_n^{(k)}}{7} \\ M_3^{(k)} &= \frac{\sum_{n=1}^7 \rho_n^{(k)}}{7}, & M_1^{(k+1)} &= \frac{\sum_{n=1}^8 \rho_n^{(k+1)}}{8} \\ M_2^{(k+1)} &= \frac{\sum_{n=2}^8 \rho_n^{(k+1)}}{7}, & M_3^{(k+1)} &= \frac{\sum_{n=1}^7 \rho_n^{(k+1)}}{7}.\end{aligned}$$

- ii) Subtract the mean $M_v^{(k)}$ from the vector $\beta_v^{(k)}$ and $M_v^{(k+1)}$ from $\beta_v^{(k+1)}$

$$\begin{aligned}\hat{\beta}_1^{(k)} &= (\rho_1^{(k)} - M_1^{(k)}, \dots, \rho_8^{(k)} - M_1^{(k)}) \\ \hat{\beta}_2^{(k)} &= (\rho_2^{(k)} - M_2^{(k)}, \dots, \rho_8^{(k)} - M_2^{(k)}) \\ \hat{\beta}_3^{(k)} &= (\rho_1^{(k)} - M_3^{(k)}, \dots, \rho_7^{(k)} - M_3^{(k)}) \\ \hat{\beta}_1^{(k+1)} &= (\rho_1^{(k+1)} - M_1^{(k+1)}, \dots, \rho_8^{(k+1)} - M_1^{(k+1)}) \\ \hat{\beta}_2^{(k+1)} &= (\rho_2^{(k+1)} - M_2^{(k+1)}, \dots, \rho_8^{(k+1)} - M_2^{(k+1)}) \\ \hat{\beta}_3^{(k+1)} &= (\rho_1^{(k+1)} - M_3^{(k+1)}, \dots, \rho_7^{(k+1)} - M_3^{(k+1)}).\end{aligned}$$

- iii) Calculate the mean square difference of $\hat{\beta}_v^{(k)}$ and $\hat{\beta}_v^{(k+1)}$ as $\Omega_v = (1/t)(\hat{\beta}_v^{(k)} - \hat{\beta}_v^{(k+1)})^2$ where $t = 8$ when $v = 1$, and $t = 7$ otherwise.

- iv) Find $\min_v \Omega_v$ and the corresponding $\beta_v^{(k)}$ and $\beta_v^{(k+1)}$.

To calculate the dc value, assuming that the k is the reference block and the pixels in the $(k+1)$ -th block are adjusted, the adjustment value Δ_{k+1} is given by $\Delta_{k+1} = (M_v^{(k)} - M_v^{(k+1)})N^2$ and the new pixel values $\rho_i^{(k+1)}$ are calculated as $\rho_i^{(k+1)} = \rho_i^{(k+1)} + \Delta_{k+1}/N^2$.

D. Bounding the dc Value of a Block

Property 2 can be used to bound the range $\Lambda^{(k)}$ of the dc coefficient of a block. Let the pixels $\rho^{(k)}$ in block $T^{(k)}$ have the range

$$\lambda_{\min}^{(k)} \leq \rho^{(k)} \leq \lambda_{\max}^{(k)} \quad (6)$$

and assume possible values of pixels are in the interval, $0 \leq \rho_i^{(k)} \leq t_{\max}, \forall k, i$. Then the following must hold:

$$0 \leq \rho^{(k)} + 1/N^2 c_{1,1}^{(k)} \leq t_{\max}. \quad (7)$$

From (6) and (7) $0 \leq \lambda_{\min}^{(k)} + 1/N^2 c_{1,1}^{(k)}$ and $\lambda_{\max}^{(k)} + 1/N^2 c_{1,1}^{(k)} \leq t_{\max}$. Hence

$$\frac{\lambda_{\min}^{(i)}}{1/N^2} \leq c_{1,1}^{(k)} \leq \frac{t_{\max} - \lambda_{\max}^{(i)}}{1/N^2}. \quad (8)$$

For an 8-bit image and a block of 8×8 pixels, the value of $t_{\max} = 255$, $N = 8$ and the values of $\lambda_{\min}^{(k)}$ and $\lambda_{\max}^{(k)}$ are fixed for a given image block k . Hence, the range of $c_{1,1}^{(k)}$ is appropriately determined.

E. Recovering the DC Value of the Image

Using the result of Sections II-C and II-D, we describe an algorithm that recovers dc signal value of blocks. The two steps of the algorithm, that is estimating relative values of dc signals and then estimating the actual dc signal value, are described in Sections II-E1 and II-E2, respectively.

1) *Adjusting Relative Values of dc Signals*: We use the methods described in Section II-C to estimate the relative dc signal values of blocks in an image in terms of their adjacent blocks. If the dc signal values of all blocks are unknown, without loss of generality we assume the top left block in the image is the reference block. The range of dc signal value for the block can be obtained from (8). We calculate the dc signals of all other blocks in terms of the dc signal of the reference block.

We note that in order to use the algorithm in Section II-C to find an estimate for $[X]_{dc}^{(j)}$, we may choose one of the four possible adjacent blocks. This means that to cover all blocks in the image starting from a reference block, various path through the image blocks can be considered.

As noted earlier, to estimate dc value of a block one or more of its neighboring blocks can be used. The algorithm below is an example of systematically adjusting all blocks of an image.

- i) First scan:

- a) The block in the upper left corner of the image is chosen as the reference block. Blocks in the first row are considered from left to right, and in each case the dc value is adjusted with respect to its left block.
- b) The rows below the first are scanned similar to the above but each block, except the left-most ones, is compared to its upper and left blocks and is adjusted based on the average of the two estimated adjustment values. For a left-most block, only its upper block is considered.

- ii) Second, third, and fourth scan:

All blocks are scanned similarly to the first scan but the scan starts from the upper right corner, the bottom left corner, the

Fig. 3. Images recovered by the method in Section II-C.

TABLE II
PSNR OF RECOVERED IMAGES WITH RESPECT TO THE ORIGINAL IMAGES

(1) Using horizontal and vertical relationship of pixels.

Image	PSNR		
	JPEG 25%	JPEG 50%	JPEG 75%
airfield256x256.pgm	24.5 dB	24.4 dB	24.1 dB
mandrill.pgm	19.7 dB	20.7 dB	20.3 dB
lena.pgm	21.3 dB	21.6 dB	24.1 dB
peppers.pgm	20.6 dB	19.8 dB	18.8 dB

(2) Using horizontal, vertical and diagonal relationship of pixels.

Image	PSNR		
	JPEG 25%	JPEG 50%	JPEG 75%
airfield256x256.pgm	24.8 dB	24.8 dB	24.4 dB
mandrill.pgm	20.0 dB	21.2 dB	21.2 dB
lena.pgm	21.4 dB	22.0 dB	25.2 dB
peppers.pgm	21.5 dB	20.8 dB	18.9 dB

bottom right corner of the image for second, third and fourth scans, respectively.

2) *Adjustment of the Range of a Pixel Value:* After the relative adjustment of the dc values of all blocks, it is necessary to find the actual values of the dc signal for the entire image. The adjustment in Section II-E1 does not take into account possible range of pixels in a block and so during the adjustment some pixels in the image may move outside the valid pixel range.

The range of dc signal value in each block can be obtained from (8). The effective range of $c_{1,1}^{(j)}$, the dc value of the reference block, is the smallest range of all blocks. This is because changing $c_{1,1}^{(1)}$ from 0 to $\alpha > 0$, adds the same value to all $c_{1,1}^{(j)}$ for all $j = 2, \dots$, and the new value of $c_{1,1}^{(j)}$ must stay within the range $\Lambda^{(j)}$.

The range of the pixel value may be larger than the valid pixel range due to the inaccuracy in the recovery. To fit all pixel values within the valid range, either all pixel values must be scaled or only the pixel values outside the valid range must be adjusted.

III. EXPERIMENT RESULTS

We used the algorithms described above to recover the images whose DCT coefficients excluding the dc coefficient are given. We used JPEG

prior to recovering dc coefficients to observe the impact of quantization on the recovery. The steps used for the experiments in Sections III-A, III-B are as follows.

- Transform the image using 8×8 two-dimensional (2-D) DCT.
- Compress and then decompress the image via JPEG (i.e., quantize the DCT coefficients) using `cjpeg` and `djpeg` commands [18]. Then the following dc recovery procedure was performed on the decompressed images.
- All the dc coefficients were set to a constant value (1023 which is the median value of the range of a dc value).
- The methods in Section II-C, II-E1 and II-E2, were used to recover the dc values.

The following tables summarize the recovery results for the two algorithms described in Section IIE, i.e., the one which only considers horizontal and vertical pixels and the one which also considers the diagonal pixels.

A. Recovery Using Horizontal and Vertical Pixels

The results show block artifacts in the active regions but in the smooth regions, there is few noticeable block artifacts. The PSNRs are shown in Table II (1) and the recovered images are shown in Fig. 3 (1). Although the PSNRs are not very good, the images are comprehensible. The impact of quantization on the recovery is insignificant. This is reasonable because setting a lower JPEG quality level results in reduction of high frequency components and so the resulting image tends to be smooth and the recovery will work better on the smooth images. Part of the decrease in PSNR for lower quality would be caused by the JPEG compression.

B. Recovery Using Horizontal, Vertical, and Diagonal Pixels

The PSNRs are shown in Table II (2) and the recovered images are shown in Fig. 3 (2). Compared with the method in Section II-C, there was a small improvement in PSNR figure. The results do not show large difference from those using the method in Section II-C.

C. Recovery Without Quantization of ac Coefficients

We also conducted the experiments in which there was no coefficient quantization. For these experiments, to hide the original dc values the

TABLE III
PSNR OF RECOVERED IMAGES WITH RESPECT TO THE ORIGINAL
IMAGES (NO COEFFICIENT QUANTIZATION)

(1) Using horizontal and vertical relationship of pixels.

Image	PSNR
airfield256x256.pgm	23.1 dB
mandrill.pgm	18.2 dB
lena.pgm	22.9 dB
peppers.pgm	17.7 dB

(2) Using horizontal, vertical and diagonal relationship of pixels.

Image	PSNR
airfield256x256.pgm	23.5 dB
mandrill.pgm	17.7 dB
lena.pgm	23.8 dB
peppers.pgm	18.6 dB

original images were transformed by 8×8 DCT and all the dc coefficients were set to 1023 (the median value of the dynamic range of dc) and then they were inverse-transformed. Then 1) using the horizontal and vertical relationship of pixels, and 2) using the horizontal, vertical, and diagonal relationship of pixels, the dc values of those images were recovered. The PSNRs of the recovered images with respect to the original images are shown in Table III.

By using the diagonal relationship in addition to the horizontal and vertical relationships, all the images except `mandrill.pgm` had improvement in their PSNRs. Table III summarizes the results. Compared with the results obtained when the images were JPEG-compressed, these PSNRs are lower. We attribute this result to the fact that the method of dc recovery we presented exploits the smoothness of images and the JPEG compression removes the higher frequency components thus making the images smoother. Without quantization, the amount of smoothness is reduced.

IV. AN ATTACK ON MPEG ENCRYPTION SCHEME

In the MPEG encryption method proposed by Tang [12], the ac coefficients in each block are permuted and the permutation is secret. Dc coefficients can be encrypted using a block cipher algorithm such as DES. Typically the same permutation is used for a frame or a several number of frames. In the decoder, the permuted ac coefficients are inverse-permuted and the pixels are recovered by inverse-DCT.

An attacker is able to find the permutation by constructing a specially designed MPEG stream. In the stream, each DCT block contains a set of distinct values of ac coefficients. The stream is passed to the decoder and the attacker is able to obtain the inverse-permuted ac coefficients by using DCT on the decoded frames. For a DCT block, attacker may not be able to choose 63 values which remain distinct through quantization. However, typically the number of blocks in a frame is large and so it is possible to assign small number of distinct values to each block such that each block may reveal only the permutation of several number of frequencies but all blocks cover the entire 63 frequencies. By applying the secret permutation to decrypt the encrypted frames, the ac coefficients can be recovered [15].

Now that attacker has a way of recovering the ac coefficients from the permutation, it remains to recover the dc coefficients. Recovering the dc coefficients which are encrypted using a block cipher can be difficult or it can be infeasible. The dc recovery method presented in this paper makes it practical to recover the dc coefficients from the known ac coefficients.

V. CONCLUSION

We showed that if block-based DCT is used on images, then it is possible to find an estimate of the dc signal value of a block from the ac

signal of that block and the complete signal of its neighboring blocks. The method selects the smoothest direction of natural images and only considers horizontal, vertical or diagonal direction. It is possible to increase the number of directions, for example, using every 45° direction, to obtain a more precise direction of smoothness and hence a better estimate of the dc signal value.

It has been argued that DCT encryption systems that use permutation of the ac coefficients together with cryptographic encryption of the dc coefficients provide high security and result in incomprehensible images. Using the attack in [15] together with the results in this paper shows that the claimed level of security does not hold.

REFERENCES

- [1] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. C-23, no. 1, pp. 90–93, Jan. 1974.
- [2] G. Strang, "The discrete cosine transform," *SIAM Rev.*, vol. 41, no. 1, pp. 135–147, 1999.
- [3] ITU, JPEG Standard: CCITT Recommendation T.81 International Telecommunication Union, 1993 [Online]. Available: <http://icib.igd.fhg.de/icib/telecom/ccitt/rec/t.81/scan.html>
- [4] MPEG-2 Encoder/Decoder MPEG Software Simulation Group, 1996 [Online]. Available: <http://www.mpeg.org/MPEG/MSSG/>
- [5] J. Meyer and F. Gadegast, *Security Mechanisms for Multimedia-Data with the Example MPEG-I-Video. Project description of SEC MPEG*. Berlin, Germany: Technical Univ. Berlin, May 1995.
- [6] Y. Li, Z. Chen, S.-M. Tan, and R. H. Campbell, "Security enhanced MPEG player," in *Proc. IEEE 1st Int. Workshop on Multimedia Software Development (MMSD 96)*, Mar. 1996, pp. 169–175.
- [7] H. Cheng and X. Li, "On the application of image decomposition to image compression and encryption," *Commun. Multimedia Security II*, pp. 116–127, 1996.
- [8] L. Qiao and K. Nahrstedt, "A new algorithm for MPEG video encryption," presented at the 1st Int. Conf. Imaging Science, Systems, and Technology, Jul. 1997.
- [9] C. Shi and B. Bhargava, "A fast MPEG video encryption algorithm," in *Proc. 6th ACM Int. Multimedia Conf.*, Sep. 1998, pp. 81–88.
- [10] —, "An efficient MPEG video encryption algorithm," in *Proc. 17th IEEE Symp. Reliable Distributed Systems*, Oct 1998, pp. 381–386.
- [11] C. Shi, S.-Y. Wang, and B. Bhargava, "MPEG video encryption in real-time using secret key cryptography," in *Proc. Int. Conf. Parallel and Distributed Processing Techniques and Applications*, Las Vegas, NV, Jun. 28–Jul. 1, 1999, pp. 2822–2828.
- [12] L. Tang, "Methods for encrypting and decrypting MPEG video data efficiently," in *Proc. ACM Multimedia*, Nov 1996, pp. 219–229.
- [13] S. U. Shin, K. S. Sim, and K. H. Rhee, "A secrecy scheme for MPEG video data using the joining of compression and encryption," in *Proc. ISW*, 1999, pp. 191–201.
- [14] L. Qiao, K. Nahrstedt, and M.-C. Tam, "Is mpeg encryption using random lists instead of zig zag order secure?," presented at the IEEE Int. Symp. Consumer Electronics, Dec 1997.
- [15] T. Uehara and R. Safavi-Naini, "Chosen DCT coefficients attack on MPEG encryption schemes," in *Proc. IEEE Pacific-Rim Conf. Multimedia*, Dec. 2000, pp. 316–319.
- [16] R. J. Clarke, *Transform Coding of Images*. London, U.K.: Academic, 1985.
- [17] J. Jiang and G. Feng, "The spatial relationship of DCT coefficients between a block and its sub-blocks," *IEEE Trans. Signal Process.*, vol. 50, no. 5, pp. 1160–1169, May 2002.
- [18] JPEG Software Release 6b (jpeg-6b) The Independent JPEG Group, 1998 [Online]. Available: <http://www.iijg.org>