Journal of Information Hiding and Multimedia Signal Processing ©2014 ISSN 2073-4212
Ubiquitous International Volume **5**, Number **3**, July **2014**

Reversible Data Embedding for MPEG-4 Video Using Middle Frequency DCT Coefficients

Sagar Gujjunoori

Department of Computer Science and Engineering
National Institute of Technology Warangal
Warangal, Andhra Pradesh 506004, INDIA
gsr@nitw.ac.in

B. B. Amberker

Department of Computer Science and Engineering
National Institute of Technology Warangal
Warangal, Andhra Pradesh 506004, INDIA
bba@nitw.ac.in

Received June, 2013; revised January, 2014

Abstract. *The advancement of Internet technology made the usage and distribution of multimedia contents over Internet very simple. Digital watermarking and steganography are becoming the areas of interest for providing security to the multimedia contents. Recently, the data embedding into multimedia contents such as image, video, audio, etc. is fascinating in watermarking and steganography applications. Designing the data embedding schemes for video and achieving reversibility become a challenging task. In this article, we propose two reversible data embedding schemes for video, which embed the data during the process of MPEG-4 compression. We use the middle frequency components of the DCT blocks for embedding. Compared to C. C. Chang et al. scheme, proposed MIDCAP achieve almost twice the embedding capacity and the acceptable range of PSNR values and MIDVIS is aimed at improving the visual quality for the applications which requires good visual quality.*
Keywords: Reversible data embedding; MPEG-4; DCT; embedding capacity; visual quality; middle frequency coefficients

---

1. **Introduction.** Data embedding into the multimedia content such as image, audio, video, etc. is attracting for various applications of steganography and watermarking due to the rapid growth of Internet technology. The watermarking and steganography contribute different kinds of most remarkable methods of embedding important data in an irremovable and/or undetectable way in the multimedia contents. A data embedding scheme alters the cover contents such as image, audio, video, etc. for embedding the data. The data to be embedded can be secret information, identity of the content owner, information about the cover content, etc. depending up on the application for which the embedding scheme is designed.

In a generic steganography (watermarking) scheme, an *Embedder* $\mathbb{E}$ is used to embed the *data* $\mathbb{I}$ (e.g., either the secret message or the watermark) into a *Cover-object* $\mathbb{C}$. The *Detector* $\mathbb{D}$ is used to detect (extract) the data from the *stego-object* (watermarked object) $\bar{\mathbb{C}}$. Though, both the steganography and watermarking schemes alter the cover content for embedding the data, their primary goals are aimed at different purposes. The main purpose of a steganography scheme is to embed the data $\mathbb{I}$ into the multimedia content $\mathbb{C}$ to obtain the new content $\bar{\mathbb{C}}$, practically indistinguishable from $\mathbb{C}$, by human beings, in such a way that the unauthorized person can not detect the existence of $\mathbb{I}$ in $\bar{\mathbb{C}}$. Whereas, the main purpose of a watermarking scheme is that the unauthorized person should not be able to remove or replace the $\mathbb{I}$ in $\bar{\mathbb{C}}$ [5].

The data embedding schemes can be classified into *irreversible* and *reversible* schemes. The design goal of a *irreversible* scheme is to embed and extract the data such that the distortion to the cover content is minimal. Here, the cover content can not be recovered perfectly. Whereas, there are some applications such as remote sensing, military image processing, fine arts, medical image sharing, multimedia archive management etc. in which the perfect recovery of the cover content is desired. The *reversible* data embedding schemes are designed to embed and extract the data by perfectly recovering the cover content [16].

Though both the *irreversible* and *reversible* data embedding schemes are explored in the literature in both the spatial domain and frequency domain, the *reversible* schemes in the frequency domain are widely focused due to its special requirement in robustness aimed applications. There are two trends in data embedding into the frequency domain. First, transforming the pixel values in the spatial domain to a particular frequency domain and then embedding the data in the transformed domain. The other trend is to make use of a compression standard for embedding into the frequency domain.

The compression standards such as JPEG, JVT, MPEG, etc. are becoming more popular in compressing the redundant data in images, videos, etc. These standards help in efficient storage and transmission of data [7]. These standards, in nut shell, transform the pixels in spatial domain to the frequency domain, quantize the transformed coefficients followed by encoding in order to compress the data. We briefly discus some of the reversible data embedding schemes which makes use of the compression standard for embedding the information into the frequency domain. Especially, we discus the reversible data embedding schemes which works on *DCT* for the matter of concern.

The DCT transforms the pixel values of an image/video sequence into varying frequency sub-band coefficients. This sequence considered as a block of a DCT consists of three regions of frequency sub-band coefficients referred to as low frequency, middle frequency and high frequency. Majority of the energy of a block resides in few low frequency DCT coefficients. The frequency coefficients in the high frequency region represents a little energy of a block which can be neglected.

There exist a few reversible schemes which modify the DCT coefficients for embedding the data. Y. Shang [19] has given an invertible data embedding scheme for compressed videos or images. G. Xuan et al. [21] have proposed a lossless data hiding scheme, which embeds the data into quantized DCT coefficients of JPEG image based on histogram pairs to achieve higher embedding capacity. H. Sakai et al. [17] have improved the Xuan et al.'s method by selecting the smoother part of JPEG image for embedding to increase the image quality and reduce JPEG file size. Recently, Q. Li et al. [12] gave a reversible data hiding scheme which works directly on quantized DCT entropy coded coefficients of JPEG image. W-C. Kuo et al. [11] have proposed a secure and high capacity reversible data hiding scheme by ensuring that the size of stego-image does not increase. C-C. Chen and D-S. Kao [2] have proposed a reversible watermarking scheme for an image which works on the quantized DCT coefficients. In this scheme, they embed one bit into a pre-defined selection of coefficients.

J. Fridrich and R. Du [6] extended the scheme of F. Jessica et al. scheme [10] to MPEG-2 video, to detect the temporal tampering in the compressed domain for video. J. Y. Park et al. [15] proposed an invertible semi-fragile video watermarking scheme for distinguishing the MPEG-2 compression from malicious manipulation. F. Gui et al. [8] have proposed a blind video watermarking scheme for MPEG-2, based on virtual channel technique to achieve good robustness and imperceptibility. S. D. Lin et al. [14] proposed an invertible data embedding scheme for error resilience in H264/AVC. H. Chen et al. [3] proposed semi-fragile video watermarking scheme to authenticate MPEG-4 video contents. They embed the watermark into the quantized DCT coefficients of a luminance component (Y) of I-$frame$.

The modifications in the low frequency region for embedding the data result in degradation of visual quality. To avoid this, the middle frequency coefficients are utilized for embedding the data. C. C. Chang et al. [1] have proposed a reversible data embedding scheme using the middle frequency coefficients of a DCT block to improve the visual quality. C-Y. Lin et al. [13] have proposed a high embedding capacity multiple-base notation approach, which embeds the data in the zero value quantized DCT coefficients which are closer to middle-frequency components. But it has the poor visual quality. Later, to improve the visual quality of stego-image and maintain the high embedding capacity, Z. Cheng and K-Y. Yoo [4] proposed a JPEG-to-JPEG data hiding method. In this method, each 8×8 DCT quantized block is divided into 3 parts: DC coefficient Block (DCB), Auxiliary Block (AB) and Data Block (DB). The secret data and auxiliary order sequence are embedded into DB and AB. X. Zeng et al. [23] proposed a lossless drift compensation scheme to reduce the distortion in P- and B-$frames$, which is due to data hiding in I-$frames$. They used C. C. Chang et al. [1] and Z. Xiao et al. [20] reversible data hiding schemes for embedding and extracting.

By observing most of the standard QCIF formatted videos, we identified that there is more amount of information in a frame which is perceptibly less significant than in a still image. This less perceptible information is eliminated by quantization during the MPEG compression as shown in FIGURE 1. This leads to large number of zeros in the high frequency region of a quantized DCT block of a video frame. By exploring this observation and inspired by C. C. Chang et al. scheme [1], we have proposed two reversible data embedding schemes for video during the MPEG-4 compression. The first scheme referred to as MIDCAP is proposed to improve the embedding capacity, and the second scheme referred to as MIDVIS is aimed at improving the visual quality.

The paper is organized as follows. Section 2 briefly reviews the MPEG-4 compression and our proposed schemes for embedding the data during the process of MPEG-4 compression will be detailed. Results and discussion is given in Section 3. We conclude the paper in Section 4.

2. **Proposed Schemes.** We propose two data embedding schemes referred to as MIDCAP and MIDVIS. These schemes use the middle frequency coefficients of quantized DCT to embed the data during the MPEG-4 compression. The MIDCAP is proposed to improve the embedding capacity. The MIDVIS scheme is aimed at improving the visual quality by embedding the data into non-smoother blocks. The MIDCAP scheme is presented using the data embedding, data extraction and data restoration procedures in the following sections 2.3, 2.4 and 2.5 respectively. The MIDVIS uses the same procedures for embedding, extraction and restoration except it embeds the data into non-smoother blocks as defined using equation (7).

2.1. **The Framework of Embedding in MPEG-4.** We embed the data into the MPEG-4 video during the process of compressing the raw YUV video into MPEG-4 format [9]. Broadly, the embedding framework in MPEG-4 include the formation of intra coded frames and inter coded frames followed by encoding. Specifically, it includes the components like DCT, quantization, embedding, prediction, encoding as shown in FIGURE 1. The MPEG-4 compression involves the formation of a sequence of three kinds of frames: I-, P-, B-$frame$. The I-$frames$ are called *reference frames* and P-, B-$frames$ are called *predicted frames*. The I-$frames$ are coded using *Intraframe* technique, i.e, they are reconstructed without having the reference to any other frames. The P-$frames$ are coded using *Interframe* technique called *forward prediction*. They are forward predicted from the recent I-$frame$ or P-$frame$. The B-$frames$ are also coded using *Interframe* technique but they are both *forward predicted* from the recent and *backward predicted* from the future I-$frame$ or P-$frame$, i.e, two other frames are necessary to reconstruct the B-$frames$. Hence, in the MPEG-4 compression the I-$frames$ are the key frames without which the reconstruction of the compressed video is not possible. Multiple feedbacks can be used by the encoder in predictive coding to improve the performance of coding. In this article, we choose the luminance component (Y) of the every I-$frame$ for embedding the data. We take the 8×8 block of a luminance component (Y) of an I-$frame$, get the quantized DCT coefficients and embed the data into it. Note that we present only the steps of interest in MPEG-4 compression in FIGURE 1.

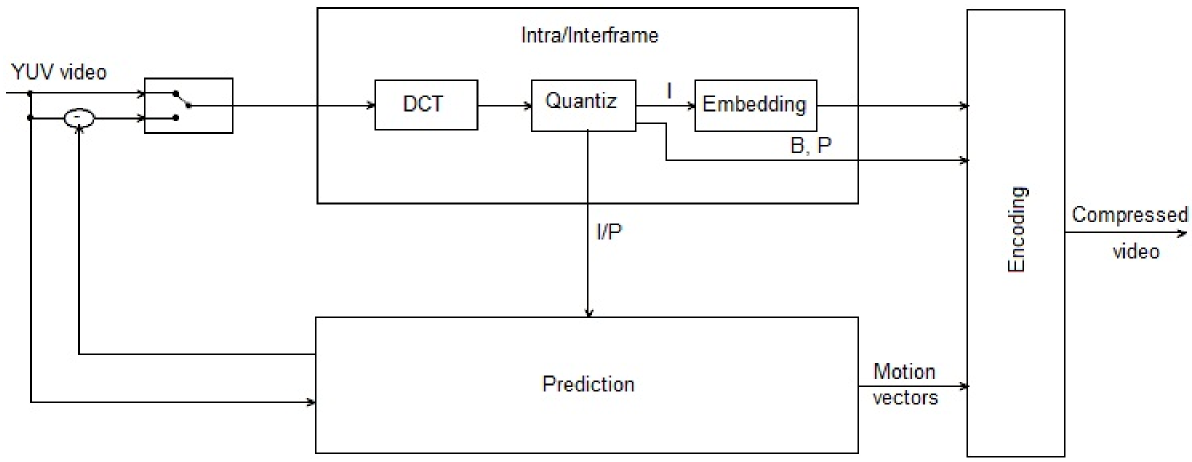


FIGURE 1. The framework of embedding in MPEG-4

2.2. **Models and Notations.** The raw YUV video consist of a sequence of frames. Let $F = \{\bar{f}_1, \bar{f}_2, ..., \bar{f}_n\}$ be the sequence of frames of raw YUV video, where n is the total number of frames [9]. Each frame $\bar{f}_i \in F$ consists of one *luminance*, two *chroma* components. Let $\bar{f}_i = \{Y, C_b, C_r\}$, where Y is the *luminance* component and C_b, C_r are the two *chroma* components of $\bar{f}_i$. All these components can be compressed using MPEG-4 encoder. The sequence of frames F is given as input to the MPEG-4 encoder. While the compression process is being carried out, the MPEG-4 encoder expresses the frames in F as the sequence of I-, P-, B-$frames$. Then $\bar{F} = I \cup P \cup B$, where I is the set of I-$frames$ called *reference frames* and P, B are the sets of P-, B-$frames$, which are the *predicted frames*.

Though all the frames in $\bar{F}$ can be used for embedding the data, we use only I-$frames$ for embedding. Let $I = \{I_1, I_2, ..., I_m\}$, where $m < n$. As we are concerned with I, let $I_i = \{Y^i, C_b^i, C_r^i\}$, where Y^i is the *luminance* component of I_i and C_b^i, C_r^i are the two *chroma* components of I_i. We consider Y^i of I_i for embedding the data. Here each Y^i, of size $n_1 \times n_2$, is partitioned into 8×8 blocks of intensity values. We assume that both n_1 and n_2 are the multiples of 8. Let $Y^i = \{B_1^i, B_2^i, \cdots, B_l^i\}$, where B_j^i is the j^{th} 8×8 block of Y^i and $l = \frac{n_1 \times n_2}{64}$. Here $\hat{m} = m \times l$ gives the total number of blocks in the set I.

These 8×8 non-overlapping blocks are transformed into 2-dimensional Discrete Cosine Transform (DCT) as follows.

$$F_{u,v} = \frac{\alpha(u)\alpha(v)}{4} \sum_{x=0}^{7} \sum_{y=0}^{7} B_j^i(x,y)\hat{g}(x,y,u,v) \tag{1}$$

where

$$\hat{g}(x,y,u,v) = \cos\left(\frac{(2x+1)u\pi}{16}\right) \cos\left(\frac{(2y+1)v\pi}{16}\right)$$

$$\alpha(e) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } e = 0, \\ 1 & \text{if } e \neq 0. \end{cases}$$

Here, $0 \leq u, v \leq 7$ and $B_j^i(x,y)$ represent the intensity value (pixel value) of block B_j^i at the coordinate (x,y) in the spacial domain and $F_{u,v}$ represent the coefficient at the coordinate (u,v) in the frequency domain. The inverse DCT ($IDCT$) is obtained by equation (2) as follows, where $\alpha(e)$, $\hat{g}$ are the same as in equation (1) and $0 \leq x, y \leq 7$.

$$B_j^i(x,y) = \sum_{u=0}^{7} \sum_{v=0}^{7} \frac{\alpha(u)\alpha(v)}{4} F_{u,v}\hat{g}(x,y,u,v) \tag{2}$$

Let $\hat{B}^i = \{\hat{B}_1, \hat{B}_2, \cdots, \hat{B}_l\}$ be the set of 8×8 blocks of DCT coefficients of Y^i, and Q be a 8×8 block of the quantization table used in *Intraframe* coding. Let $C^i = \{C_1, C_2, \cdots, C_l\}$ be the set of 8×8 blocks of quantized DCT coefficients and $\bar{C}^i = \{\bar{C}_1, \bar{C}_2, \cdots, \bar{C}_l\}$ be the set of embedded blocks of Y^i. Let $\mathbb{C} = \{C^i\}$ be the set of elements of video sequence to be considered for embedding. Let D_k $(1 \leq k \leq 9)$ be the set of quantized DCT coefficients from high frequency to low frequency of a 8×8 block as shown in FIGURE 2. We use these sets for embedding the data. Let $(d_{k,1}, d_{k,2}, \ldots, d_{k,K(k)})$ be the sequence of quantized DCT coefficients in the set D_k in C^i, where $K(k)$ is given in TABLE 1.

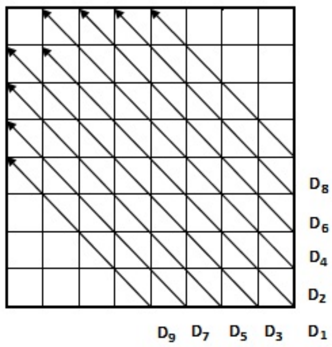


FIGURE 2. Chosen sets for embedding

TABLE 1. The size of the chosen sets for embedding

k	1	2	3	4	5	6	7	8	9
$K(k)$	7	7	7	6	6	5	5	4	4

2.3. **Data Embedding Procedure.** For embedding the data, we choose the middle frequency components as in C.C.Chang scheme [1]. We consider the sets D_k for embedding as shown in FIGURE 2. In each set D_k, we embed two consecutive data bits $(s, t) \in \mathbb{I}$, if it is *suitable* for embedding. We decide the suitability of a set D_k for embedding based on the number of ceaseless zeros from high frequency to low frequency in the set. Let b_k be the number of ceaseless zeros of the set D_k from high frequency to low frequency. If $b_k \geq 3$, then the set D_k is *suitable* for embedding and we choose this set for embedding.

In general, to make a data embedding scheme to be reversible, one should be able to extract the embedded data exactly as it is and should be able to invert the modified object after embedding the data, back to its original form. In the process of achieving the reversibility of a scheme, the most commonly originated issues such as *data collusion* and *overflow* need to be resolved [22]. The *data collusion* refers to the situation where the altered data (for instance, pixel values of a frame, transformed coefficients, etc.) as a result of data embedding is confused with unaltered data during the data extraction and original image/video recovery. The *overflow* refers to the situation where the coefficient (pixel) values of the image/video exceed the allowed range of upper or lower bounds.

In the proposed scheme, the problem of overflow is avoided due to the quantization of DCT coefficients, since the quantization restricts the relatively large real numbers of a DCT block to a relatively small integers. We alter the quantized DCT coefficients only by ± 1 to achieve the reversibility. The alteration by ± 1 is very small. Hence, the alteration of quantized DCT coefficients by ± 1 would not give rise to the problem of overflow in the DCT coefficients. To resolve the data collusion issue, we define several *ambiguous conditions* which lead to the data collusion issue, and its remedial measures. Let $d_{k,j}$ be the first non-zero element if it exists in D_k, which leads to data collusion issue in the following ambiguous conditions for $1 \leq k \leq 9$ and $1 \leq j \leq K(k)$.

The *ambiguous conditions* are as follows:

- **(A):** *If* $b_k < 3$, *and* $d_{k,j} \neq 0, d_{k,j+1} = d_{k,j+2} = 0$, *for* $j = 1$ *and* $j + 2 \leq K(k)$.
- **(B):** *If* $b_k < 3$, *and* $d_{k,j} \neq 0, d_{k,j+1} = \pm 1$, $d_{k,j+2} = 0$, *for* $1 \leq j \leq 3$ *and* $j + 2 \leq K(k)$.
- **(C):** *If* $b_k < 3$, *and* $d_{k,j} \neq 0, d_{k,j+1} = 0$, *for* $2 \leq j \leq 3$ *and* $j + 1 \leq K(k)$.
- **(D):** *If* $3 \leq b_k \leq 5$, *and* $d_{k,j} \neq 0, d_{k,j+1} = 0$ *for* $j + 1 \leq K(k)$ *and the data bits are* $s = 0, t = 0$.
- **(E):** *If* $3 \leq b_k \leq 4$, *and* $d_{k,j} \neq 0, d_{k,j+1} = \pm 1, d_{k,j+2} = 0$ *for* $j + 2 \leq K(k)$, *and the data bits are* $s = 0, t = 0$.

Before embedding, if any of the ambiguous conditions from (**A**) to (**E**) holds in the set D_k, then we modify the value of $d_{k,j} \neq 0$ to $d'_{k,j}$ to eliminate the data collusion issue as in equation (3).

$$d'_{k,j} = \begin{cases} d_{k,j} + 1, & \text{when } d_{k,j} > 0, \\ d_{k,j} - 1, & \text{when } d_{k,j} < 0. \end{cases} \tag{3}$$

After eliminating the data collusion problem, we embed the two data bits s and t into the suitable set D_k as follows. When $D_k = (d_{k,1}, d_{k,2}, \ldots, d_{k,K(k)}) = (0, 0, \ldots, 0)$, then let $z_{k,1}$ be the first zero value, $z_{k,2}$ be the second zero value, and $z_{k,3}$ be the third zero value from low frequency to high frequency in the set D_k. And let $z_{k,1}$, $z_{k,2}$, $z_{k,3}$ be the zero values which represent the values of $d_{k,j-1}$, $d_{k,j-2}$, $d_{k,j-3}$ respectively in the set D_k, for $4 \leq j \leq K(k)$, when $d_{k,j} \neq 0$. We use the values of $z_{k,3}$ and $z_{k,2}$ to indicate the data bits s and t respectively in the set D_k. Note that $z_{k,3}$ and $z_{k,2}$ will not exist if $b_k < 3$. If $b_k \geq 3$, then we modify the values $z_{k,3}$ and $z_{k,2}$ to indicate the data bits as in equations (4) and (5) respectively. The data embedding algorithm for MIDCAP scheme is given in Algorithm 1.

$$z_{k,3} = \begin{cases} 0, & \text{when } s = 0, \\ \pm 1, & \text{when } s = 1. \end{cases} \tag{4}$$

$$z_{k,2} = \begin{cases} 0, & \text{when } t = 0, \\ \pm 1, & \text{when } t = 1. \end{cases} \tag{5}$$

where $+1$ or -1 is randomly selected when $s = 1$ and $t = 1$ in the equations (4) and (5) respectively.

Algorithm 1: Data Embedding Algorithm of MIDCAP Scheme

Input : $I = \{I_1, I_2, ..., I_m\}$ be the set of I-frames and $\mathbb{I}$ be the data to be embedded.

Output: The set of $I\text{-}frames$ with embedded data.

forall the $I_i \in I$ **do**
- Extract the Y^i from I_i
- Partition $Y^i \to \{B^i_1, B^i_2, \cdots, B^i_l\}$
- **foreach** $B^i_j \in Y^i$, *where* $1 \le j \le l$ **do**
  - *Find the DCT of* B^i_j: $\hat{B}^i_j = DCT(B^i_j)$;
  - *Quantize the DCT coefficients in* $\hat{B}^i_j$ *as below*;
  - **for** $i_1 \leftarrow 1$ **to** *8* **do**
    - **for** $i_2 \leftarrow 1$ **to** *8* **do**
      - $C_j(i_1, i_2) = \hat{B}^i_j(i_1, i_2)/Q(i_1, i_2)$;
    - **end**
  - **end**
  - Consider $D_k (1 \le k \le 9)$ sets of C_j as shown in FIGURE 2;
  - Let b_k be the number of ceaseless zeros from high frequency to low frequency in the set D_k;
  - Let $d_{k,j} \neq 0$ be the first non-zero element from high frequency to low frequency if it exists in D_k;
  - **if** $((\boldsymbol{A}) \vee (\boldsymbol{B}) \vee (\boldsymbol{C}) \vee (\boldsymbol{D}) \vee (\boldsymbol{E}))$ **then**
    - *Modify* $d_{k,j}$ *as in equation (3)*;
  - **end**
  - **if** $(3 \le b_k \le 7)$ **then**
    - *Embed the data bits* $(s, t) \in \mathbb{I}$ *into* D_k *using the equations (4) and (5)* ;
  - **end**
  - Let the resultant block be $\bar{C}_j$;
- **end**
- *Combine all the* $\bar{C}_j$ *blocks into* $\bar{C}^i$: $\bar{C}^i \leftarrow \{\bar{C}_1, \bar{C}_2, \cdots, \bar{C}_l\}$;
- *Restore the* $\bar{C}^i$ *back to* $I_i = \{\bar{C}^i, C^i_b, C^i_r\}$;

end

2.4. **Data Extraction Procedure.** We extract the data based on the existence of a non-zero value from high frequency to low frequency in the set D_k. For each set D_k in the embedded block, let $d_{k,j}$ be the highest frequency non-zero component if it exists and extract the bits s and t if they present in set D_k using the rules from **Rule 1** to **Rule 12**.

Rule 1: *If* $d_{k,j-1} = 0, d_{k,j} = \pm 1, d_{k,j+1} = 0, d_{k,j+2} \neq 0$, *for* $3 < j + 2 \le K(k)$, *then* $s = 0, t = 1$ *and mark* $d_{k,j-1}$ *as* $z_{k,3}$, $d_{k,j}$ *as* $z_{k,2}$.

Rule 2: *If* $d_{k,j-2} = d_{k,j-1} = 0, d_{k,j} = \pm 1, d_{k,j+1} = 0$*, for* $3 < j+1 = K(k)$*, then* $s = 0, t = 1$ *and mark* $d_{k,j-1}$ *as* $z_{k,3}$*,* $d_{k,j}$ *as* $z_{k,2}$*.*

Rule 3: *If* $d_{k,j} = \pm 1, d_{k,j+1} = d_{k,j+2} = 0, d_{k,j+3} \neq 0$*, for* $3 < j+3 \leq K(k)$*, then* $s = 1, t = 0$ *and mark* $d_{k,j}$ *as* $z_{k,3}$*,* $d_{k,j+1}$ *as* $z_{k,2}$*.*

Rule 4 : *If* $d_{k,j-1} = 0, d_{k,j} = \pm 1, d_{k,j+1} = d_{k,j+2} = 0$*, for* $3 < j+2 = K(k)$*, then* $s = 1, t = 0$ *and mark* $d_{k,j}$ *as* $z_{k,3}$*,* $d_{k,j+1}$ *as* $z_{k,2}$*.*

Rule 5: *If* $d_{k,j} = d_{k,j+1} = \pm 1, d_{k,j+2} = 0, d_{k,j+3} \neq 0$*, for* $3 < j+3 \leq K(k)$*, then* $s = t = 1$ *and mark* $d_{k,j}$ *as* $z_{k,3}$*,* $d_{k,j+1}$ *as* $z_{k,2}$*.*

Rule 6: *If* $d_{k,j-1} = 0, d_{k,j} = d_{k,j+1} = \pm 1, d_{k,j+2} = 0$*, for* $2 < j+2 \leq K(k)$*, then* $s = t = 1$ *and mark* $d_{k,j}$ *as* $z_{k,3}$*,* $d_{k,j+1}$ *as* $z_{k,2}$*.*

Rule 7: *If* $d_{k,j-3} = d_{k,j-2} = d_{k,j-1} = 0, |d_{k,j}| > 1$*, for* $3 < j \leq K(k)$*, then* $s = t = 0$ *and mark* $d_{k,j-3}$ *as* $z_{k,3}$*,* $d_{k,j-2}$ *as* $z_{k,2}$*.*

Rule 8: *If* $d_{k,j-3} = d_{k,j-2} = d_{k,j-1} = 0, d_{k,j} = \pm 1$*, for* $j = K(k)$ *and* $j > 3$*, then* $s = t = 0$ *and mark* $d_{k,j-3}$ *as* $z_{k,3}$*,* $d_{k,j-2}$ *as* $z_{k,2}$*.*

Rule 9: *If* $d_{k,j-3} = d_{k,j-2} = d_{k,j-1} = 0, d_{k,j} = \pm 1, d_{k,j+1} \neq 0$ *for* $4 < j+1 \leq K(k)$*, then* $s = t = 0$ *and mark* $d_{k,j-3}$ *as* $z_{k,3}$*,* $d_{k,j-2}$ *as* $z_{k,2}$*.*

Rule 10: *If* $d_{k,j} \neq \pm 1, d_{k,j+1} = \pm 1, d_{k,j+2} = 0$*, for* $1 \leq j \leq 3$*, then mark none data bits exist.*

Rule 11: *If* $d_{k,j} \neq \pm 1, d_{k,j+1} = 0$*, for* $2 \leq j \leq 3$*, then mark none data bits exist.*

Rule 12: *If* $d_{k,j} \neq \pm 1, d_{k,j+1} = 0, d_{k,j+2} = 0$*, for* $j = 1$*, then mark none data bits exist.*

If $D_k = (d_{k,1}, d_{k,2}, \ldots, d_{k,K(k)}) = (0, 0, \ldots, 0)$, then use the following **Rule 13** to extract the data bits.

Rule 13: *If* $4 \leq j \leq K(k)$*, then* $s = t = 0$ *and mark* $d_{k,K(k)-2}$ *as* $z_{k,3}$*,* $d_{k,K(k)-1}$ *as* $z_{k,2}$*.*

2.5. **Restoration Procedure.** The restoring procedure starts after the extraction procedure is completed. Note that we have marked the existence of the data bits in each set D_k during the extraction procedure. If the data is embedded at $z_{k,3}$ and $z_{k,2}$ in the set D_k, first replace $z_{k,3}$ and $z_{k,2}$ with 0 and then restore the original value of the modified coefficient in each set D_k as described below. Let the $d'_{k,j}$ be the first non-zero value from high frequency to low frequency in the set D_k. We eliminate the data collusion problem by modifying $d'_{k,j}$ to $d_{k,j}$ as in equation (6), if any of the rules from **Rule A** to **Rule E** holds.

$$d_{k,j} = \begin{cases} d'_{k,j} - 1, & \text{when } d'_{k,j} > 1, \\ d'_{k,j} + 1, & \text{when } d'_{k,j} < -1. \end{cases} \tag{6}$$

Rule A: *If the data bits* s, t *exist in the set* D_k *and* $d_{k,j} \neq \pm 1, d_{k,j+1} = 0$*, for* $j+1 \leq K(k)$*.*

Rule B: *If the data bits* s, t *exist in the set* D_k *and* $d_{k,j} \neq \pm 1, d_{k,j+1} = \pm 1, d_{k,j+2} = 0$*, for* $j+2 \leq K(k)$*.*

Rule C: *If the data bits* s, t *do not exist in* D_k *and* $d_{k,j} \neq \pm 1, d_{k,j+1} = \pm 1, d_{k,j+2} = 0$*, for* $1 \leq j \leq 3$ *and* $j+2 \leq K(k)$*.*

Rule D: *If* s, t *do not exist in* D_k *and* $d_{k,j} \neq \pm 1, d_{k,j+1} = 0$*, for* $2 \leq j \leq 3$ *and* $j+1 \leq K(k)$*.*

Rule E: *If* s, t *do not exist in* D_k *and* $d_{k,j} \neq \pm 1, d_{k,j+1} = 0, d_{k,j+2} = 0$*, for* $j = 1$ *and* $j+2 \leq K(k)$*.*

The data extraction algorithm of MIDCAP scheme is given in Algorithm 2.

To improve the visual quality, we propose to embed the data into non-smoother parts of the video frame. That is, we do not alter the smoother blocks for embedding the data. For that, we need to find out the non-smoother blocks in the quantized DCT domain. The non-smoother blocks contain the non-zero AC coefficients in their respective quantized DCT blocks. We refer to these blocks as the *suitable blocks* for embedding. Let $C = \{c_1, c_2, \ldots, c_k\}$ be a quantized DCT block, where $k = 64$. We use the following equation (7) for finding out a suitable quantized DCT block for embedding, which are on the non-smoother parts of a video frame.

$$T = \sum_{i=2}^{k-2} |c_i| \tag{7}$$

The second data embedding scheme MIDVIS uses the above described data embedding, extraction, restoration procedures for embedding the data into non-smoother blocks when $T > 0$ using the equation (7).

Algorithm 2: Data Extraction Algorithm of MIDCAP Scheme

Input : I, the set of $I\text{-}frames$ with embedded data.
Output: The set of restored $I\text{-}frames$, and the extracted data: $\hat{\mathbb{I}}$.

forall the $I_i \in I$ **do**
  Extract the $\bar{C}^i$ from I_i ;
  Partition $\bar{C}^i \rightarrow \{\bar{C}_1, \bar{C}_2, \cdots, \bar{C}_l\}$;
  foreach $\bar{C}_j \in \bar{C}^i$ **do**
    Consider $D_k (1 \leq k \leq 9)$ sets of $\bar{C}_j$ as shown in FIGURE 2;
    *Extract the data bits s, t if they exist in any D_k, using the rules **Rule 1** to **Rule 13***;
    *Replace $z_{k,3}$, $z_{k,2}$ if exist in any D_k with 0 and restore the set D_k using the rules **Rule A** to **Rule E***;
    Let the resultant block be $\bar{E}_j$;
    Dequantize the elements of $\bar{E}_j$ as follows ;
    for $i_1 \leftarrow 1$ **to** *8* **do**
      for $i_2 \leftarrow 1$ **to** *8* **do**
        $R_j(i_1, i_2) = \bar{E}_j(i_1, i_2) \times Q(i_1, i_2)$;
      end
    end
    $\hat{R}_j = IDCT(R_j)$;
  end
  Combine all the $\hat{R}_j$ blocks to get the $\hat{R}^i$;
  $\hat{R}^i \leftarrow \{\hat{R}_1, \hat{R}_2, \cdots, \hat{R}_l\}$;
  Restore the $\hat{R}^i$ back to $I_i = \{\hat{R}^i, C_b^i, C_r^i\}$;
end

---

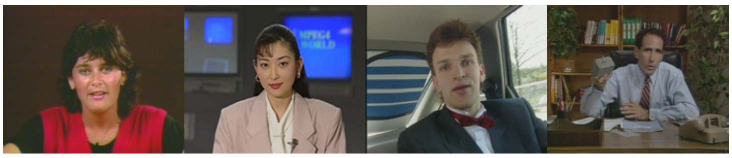
3. **Results and Discussion.** We compare our proposed scheme with the C. C. Chang et al. scheme. Note that C. C. Chang et al. scheme originally devised for images. We use various QCIF formatted videos in our experiment, including MissAm, Akiyo, CarPhone, SalesMan, etc. Some of the test videos are shown in FIGURE 3. The frame size of all these test videos is 176×144 pixels. We compress these test videos by the standard MPEG-4

encoder. The metrics PSNR and embedding capacity are used to evaluate performance of the proposed schemes. The PSNR for each YUV channel of a frame is given by the following equation.

$$\text{PSNR} = 10\log_{10}\frac{255^2}{MSE}, (dB) \tag{8}$$

where
$MSE = \dfrac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}(f_{x,y} - f^{'}_{x,y})^2$ and $f_{x,y}$, $f^{'}_{x,y}$ are the pixel values at the coordinate (x, y) of original and distorted (embedded) video YUV channels respectively, each of size $M \times N$.

We define the embedding capacity as the number of bits that can be embedded into a single Y^i. Let n be size of a single Y^i. Using C. C. Chang et al. scheme, one can embed at the maximum of $\frac{9n}{64}$ data bits, since for every 8×8 block the sets, D_k's ($1 \leq k \leq 9$) are fixed and in every set D_k only one bit can be embedded at the maximum. Using the MIDCAP and MIDVIS schemes, we can embed two data bits in every set D_k at the maximum. Hence, using our schemes we can embed $\frac{9n}{32}$ data bits at the maximum. The *ratio* between the embedding capacities of our schemes and C. C. Chang et al. scheme is given by $ratio = \frac{9n/64}{9n/32} = 2$. This *ratio* states that the proposed schemes achieve a maximum of twice the embedding capacity of C. C. Chang et al. scheme. The experimental results are shown in TABLE 2, where the $PSNR$ is calculated for the Y channel. We can observe that the $PSNR$ values with the MIDCAP and MIDVIS schemes are within the range of typical PSNR values (20-40) [18]. From the TABLE 2, it is evident that our proposed MIDCAP achieves on average a $ratio1 = 1.863$, which is almost double the embedding capacity of C. C. Chang et al. scheme for the same test videos by compromising the PSNR on average of $diff1 = 2.977$ dB. Similarly it can be observed from the TABLE 2 that the MIDVIS scheme achieve a $ratio2 = 1.656$ with $diff2 = 2.356$. FIGURE 4 shows some of the embedded videos.



(A) (B) (C) (D)

FIGURE 3. The original I frames: (A) MissAm, (B) Akiyo, (C) CarPhone, (D) SalesMan

(A) (B) (C) (D)

FIGURE 4. The embedded I frames using MIDCAP: (A) MissAm, (B) Akiyo, (C) CarPhone, (D) SalesMan

Table 2. Comparison of proposed schemes with C. C. Chang et al. scheme for various test videos

	C. C. Chang et al.		MIDCAP		MIDVIS			
Video sequence	Capacity ($cap1$)	$PSNR$ ($Q1$)	Capacity ($cap2$)	$PSNR$ ($Q2$)	Capacity ($cap3$)	$PSNR$ ($Q3$)	$ratio1 = \frac{cap2}{cap1}$	$diff1 = Q1 - Q2$
MissAm	3561	31.1126	7064	27.4253	3536	29.8469	1.984	3.6873
MotherDaughter	3559	30.3029	6998	26.6785	5414	27.5324	1.966	3.6244
Claire	3548	30.6619	6966	27.4469	3924	29.3817	1.963	3.2150
Bridge(far)	3538	30.9229	6948	27.6383	6048	28.1024	1.964	3.2846
Suzie	3547	30.3439	6936	26.9688	5982	27.5284	1.955	3.3751
Highway	3526	30.9086	6920	27.6510	5876	28.1145	1.963	3.2576
GrandMother	3534	30.1875	6882	26.7355	5064	27.7141	1.947	3.4520
Silent	3549	29.5210	6816	26.1561	6726	26.1735	1.921	3.3649
Akiyo	3525	30.2268	6806	26.9474	5582	27.4380	1.931	3.2794
Table	3521	29.9598	6728	26.7578	6566	26.8487	1.911	3.2020
CarPhone	3506	29.9575	6698	26.7883	6158	26.9040	1.910	3.1692
Foreman	3522	29.4021	6648	26.4139	5928	26.8114	1.888	2.9882
Soccer	3527	29.4872	6642	26.3430	6264	26.5710	1.883	3.1442
SalesMan	3506	29.3711	6632	26.3364	6380	26.3465	1.892	3.0347
Hall	3492	29.8099	6586	26.7950	5776	27.1634	1.886	3.0149
Coastguard	3481	29.5041	6502	26.4255	6430	26.3944	1.868	3.0786
News	3445	29.6728	6402	26.5779	5286	27.0760	1.858	3.0949
Bridge(close)	3382	29.5552	6314	26.7703	5432	27.1072	1.867	2.7849
Container	3431	29.5185	6246	26.8525	4860	27.4901	1.820	2.6660
Tempete	3431	27.9736	5748	25.6134	5676	25.6480	1.675	2.3602
Bus	3132	27.9080	5368	25.8743	5278	25.8839	1.714	2.0337
Football	2999	27.8160	4992	26.0110	4956	26.0236	1.665	1.8050
Mobile	2983	26.9453	4234	25.3757	4072	25.4984	1.419	1.5696

4. **Conclusion.** Reversibility of a data embedding scheme is the desired property in the applications such as military, imaging, etc. Achieving the reversibility, embedding more data and restraining the PSNR values in the acceptable range is a challenging task. To address these issues, we have proposed reversible data embedding schemes which embed the data into middle frequency components during the MPEG-4 compression. Our experimental results have shown that the proposed MIDCAP achieves almost double the embedding capacity of C. C. Chang et al. scheme by maintaining the acceptable visual quality of the embedded video and MIDVIS has improved the visual quality for the applications which require better visual quality. Both the proposed schemes achieve the reversibility.

References

[1] C. C. Chang, C. C. Lin, C. S. Tseng, and W. L. Tai, Reversible hiding in DCT-based compressed images, *Information Sciences: an International Journal*, vol. 177, no. 13, pp. 2768-2786.

[2] C. C. Chen, and D. S. Kao, DCT-based reversible image watermarking approach, In *Proc. of the 3rd International Conference on International Information Hiding and Multimedia Signal Processing*, pp. 489-492, 2007.

[3] H. Chen, Z. Chen, X. Zeng, W. Fan, and Z. Xiong, A novel reversible semi-fragile watermarking algorithm of MPEG-4 video for content authentication, In *Proc. of the 2nd International Symposium on Intelligent Information Technology Application*, pp. 37-41, 2008.

[4] Z. Cheng, and K. Y. Yoo, A reversible JPEG-to-JPEG data hiding technique, *Proc. of the 4th International Conference on Innovative Computing, Information and Control*, pp. 635-638, 2009.

[5] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, *Digital watermarking and steganography*, Morgan Kaufman, 2008.

[6] J. Fridrich, and R. Du, Lossless authentication of MPEG-2 video, *Proc. of International Conference on Image Processing*, pp. 893-896, 2002.

[7] B. Furht, A survey of multimedia compression techniques and standards. Part I: JPEG standard, *Real-Time Imaging*, vol. 1, no. 1, pp. 49-67, 1995.

[8] G. Feng, Q. Lin, and P. Wang, Watermarking algorithm based on virtual channel technique, *Proc. The 8th International Conference on Signal Processing*, pp. 12-20, 2006.

[9] S. Gujjunoori, and B. B. Amberker, Dct based reversible data embedding for mpeg-4 video using hvs characteristics, *Information Security Technical Report (2013).*

[10] F. Jessica, G. Miroslav, and R. Du, Invertible authentication watermark for jpeg images, *Proc. of International Conference on Information Technology: Coding and Computing*, pp. 223-227, 2001.

[11] W. C. Kuo, S. H. Kuo, and L. C. Wuu, High embedding reversible data hiding scheme for jpeg, *Proc. of the 6th International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 74-77, 2010.

[12] Q. Li, Y. Wu, and F. Bao, A reversible data hiding scheme for JPEG images, *Proc. of the 11th Pacific Rim conference on Advances in multimedia information processing: Part I*, pp. 653-664, 2010.

[13] C. Y. Lin, C. C. Chang, and Y. Z. Wang, Reversible steganographic method with high payload for JPEG images, *IEICE - Transactions on Information and Systems*, vol. E91-D, no. 3, pp. 836-845, 2008.

[14] S. D. Lin, H. C. Meng, and Y. L. Su, A novel error resilience using reversible data embedding in H.264/AVC. *Proc. of the 6th International Conference on Information, Communications and Signal Processing*, pp. 1-5, 2007.

[15] J. Y. Park, J. H. Lim, G. S. Kim, and C. S. Won, Invertible semi-fragile watermarking algorithm distinguishing MPEG-2 compression from malicious manipulation, *Proc. of International Conference on Consumer Electronics*, pp. 18-19, 2002.

[16] G. Sagar, and B. B. Amberker, A DCT based reversible data hiding scheme for mpeg-4 video, *Proc. of International Conference on Signal, Image and Video Processing*, pp. 254-259, 2012.

[17] H. Sakai, M. Kuribayashi, and M. MORII, Adaptive reversible data hiding for JPEG images, *Proc. of International Symposium on Information Theory and Its Applications*, pp. 1-6, 2008.

[18] D. Salomon, *Data Compression: The Complete Reference*, Springer-Verlag, Berlin-Heidelberg, Germany, 2007.

[19] Y. Shang, A new invertible data hiding in compressed videos or images, *Proc. of the 3rd International Conference on Natural Computation - Volume 05*, pp. 576-580, 2007.

[20] X. Zeng, Z. Chen, M. Chen, and Z. Xiong, Invertible image watermarking based on zero coefficient index, *Journal of Computer Research and Development*, vol. 47, no. 7, 2010.

[21] G. Xuan, Y. Q. Shi, Z. Ni, P. Chai, X. Cui, and X. Tong, Reversible data hiding for JPEG images based on histogram pairs, *Proc. of the 4th international conference on Image Analysis and Recognition*, pp. 715-727, 2007.

[22] G. Xuan, and Y. Q. Shi, Reversible data hiding, 2010.

[23] X. Zeng, Z. Chen, and Z. Xiong, Issues and solution on distortion drift in reversible video data hiding, *Multimedia Tools and Applications*, vol. 52, no. 2-3, pp. 465-484, 2011.