Chapter 2

Steganographic Methodologies

In this section, some tools and methodologies, used at the time of development of the proposed schemes, have been discussed.

2.1 Pixel Value Difference (PVD)

Pixel Value Difference is a novel and efficient steganographic method for secret data hiding. It can successfully provide both outstanding imperceptibility and high embedding capacity for the stego images. At first, a cover image is partitioned into non-overlapping blocks of two consecutive pixels. A difference value is then calculated from the pixel values of the two pixels in each block. All possible difference values of the pixels are classified into a number of ranges. The selection of the range intervals is based on the characteristics of the human visual system from smoothness to contrast. The difference value is then replaced by a new value to embed some secret bits. The difference value belongs to a range. The width of this range decides the number of bits which can be embedded in the pixel pair. The modification of the pixel value is never out of the range interval. The embedded secret bits can be easily extracted from the stego image without the original cover image. A pseudo-random mechanism can be used to increase the secrecy of the scheme. The PVD method can be described as follows:
The cover image is partitioned into non-overlapping blocks of two consecutive pixels p_i and p_{i+1}. From each block, a difference value $d_i = |p_i - p_{i+1}|$ can be obtained; where d_i ranges from 0 to 255. If d_i is small, the block is located within the smooth area and will embed less secret bits. Otherwise, it is located on the edge area, and it can embed a greater amount of secret bits. The quantization range table is designed with n contiguous ranges, and the range table ranges from 0 to 255. The number of secret bits hidden in two consecutive pixels depends on the quantization range table. The embedding algorithm can be described as follows:

Step 1: Calculate the difference $d_i = |p_i - p_{i+1}|$ between two consecutive pixels p_i and p_{i+1} for each block.

Step 2: Search the range table to determine how many bits will be embedded. Obtain the range R_i in which $R_i = [l_i, u_i]$, where l_i and u_i are the lower bound and the upper bound of R_i, and $m = [log_2(ui - li)]$ is the number of embedding bits.

Step 3: Read m secret bits from the secret bitstream, and transform it into decimal value b.

Step 4: Calculate the new difference $d'_i = l_i + b$. Both d_i and d'_i must be in the same range R_i.

Step 5: Assign average of p_i and p_{i+1} to d'_i. The new pixel values p'_i and p'_{i+1} are obtained by the formula given by Wu and Tsai [3].

Step 6: All the secret bits are embedded by repeating step 1 to step 5, and the stego image is generated.

In the extracting phase, the same Steps 1 and 2 in the embedding algorithm are used. The difference $d'_i = |p'_i - p'_{i+1}|$ is computed for each two consecutive pixels of the stego image and the same range table is searched to find l_i. Then, the b is computed as $b = d'_i - l_i$ and b is transformed into the binary stream. The steps are repeated until all secret data is completely extracted.

2.2 Weighted Matrix

A weighted matrix based steganographic scheme can hide multiple secret bits by changing a single bit in the cover image. A weighted matrix (WM) is simply an $(m \times n)$ integer matrix. The matrix is shared by the sender and the receiver, and it acts as a secret key. Any arbitrary values can be chosen as the element of WM from the combination of $(0, 1, \ldots, 2^{k-1}+1)$ values where k denotes the number of secret bits $(p_1 p_2 \ldots p_k)$ that can be embedded into the cover image (C). Now a new value v is generated by the following equation:

$$v = (p_1 p_2 \ldots p_k)_2 - \sum (C \otimes WM)(mod\ 2^k) \tag{2.1}$$

where $\otimes$ denotes entry-wise multiplication operator. If v is equal to zero with respect to modulo 2^k then C remains intact; otherwise, modify C to satisfy the following equation:

$$\sum (C \otimes WM) = p_1 p_2 \ldots p_k (mod\ 2^k) \tag{2.2}$$

The receiver can derive $(p_1 p_2 \ldots p_k)$ by computing $\sum (C \otimes WM)(mod 2^k)$.

2.3 Graph Neighbourhood

Some novel and efficient steganographic methods have been introduced by researchers using the image as a graph. In these schemes, images are partitioned into $(m \times n)$ sized non-overlapping

pixel blocks. Every $(m \times n)$ pixel block is considered as a separate graph where pixels are the nodes of the graph. A reference table is used to mark the group the pixels in a pixel block. A reference table contains a range of values like 0-50, 51-100, 101-150, 151-200 and 201-255. The pixel with value 30 and a pixel with value 45 will be in the same group. Whereas, a pixel with value 110 and a pixel with value 152 will be in the separate group. Each pixel in a pixel block is marked by the group number they belong. It is possible for any pixel to have another pixel having the same group number. The degree of neighbourhood of a pixel or a node is the number of pixels, belong to the same group, around it. For example, if a pixel p belongs to group 1 and there are 3 pixels around it that have the same group number 1, then the degree of neighbourhood of the pixel p will be 3. Instead of selecting all the pixels of the pixel block, a set of pixels is chosen for embedding secret data. A threshold value is used to select the pixels in the pixels block. The pixels whose degree of neighbourhood is equal or more than the threshold value is selected. These selected pixels are considered for secret data embedding. This process is demonstrated using an example. The Table 2.1 is the range table and the Table 2.2 represents the original pixel matrix. The Table 2.3 shows the group number of the pixels in the pixel block. The group number is calculated according to the range given in Table 2.1. The Table 2.4 displays the degree of neighbourhood of the pixels. For an example, the degree of neighbourhood of the last pixel in the block is 2 because there are 2 pixels around it having the same group number. Now, if the threshold value is 3, the pixels having neighbourhood degree 3 or more will be selected for data embedding. In this case, the pixels that will be selected is marked in blue in Table 2.5.

Table 2.1: *Range table used in graph neighbourhood based scheme*

Group No.	Pixel Range
1	0-49
2	52-99
3	102-149
4	152-199
5	202-255

Table 2.2: *Pixel matrix used in graph neighbourhood based scheme*

20	55	108	254
122	234	243	222
142	133	98	143
44	121	67	98

2.4 Discrete Cosine Transform (DCT)

Discrete Cosine Transform (DCT) converts a digital signal into the transform domain. The one-dimensional DCT is used for processing one-dimensional signals such as speech whereas

Table 2.3: *Group numbering of pixels according to their range in range table*

1	2	3	5
3	5	5	5
3	3	2	3
1	3	2	2

Table 2.4: *Degree of neighbourhood of pixels in a pixel block*

0	0	0	2
2	1	3	2
3	3	2	0
0	2	2	2

Table 2.5: *Selected pixels in a pixel block for a particular threshold value*

20	55	108	254
122	234	243	222
142	133	98	143
44	121	67	98

2D DCT is useful for image and video processing. In DCT, most significant information of a digital image is represented by low-frequency coefficients. This property is known as energy compaction. Because of its strong energy compaction property, it is used for lossy data compression in JPEG images. This energy compaction feature of DCT separates and removes insignificant high-frequency components from images. The two-dimensional forward transform formula of DCT is given below:

$$F(x,y) = c(x)c(y)\frac{2}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} f(u,v)cos(\frac{2u+1}{2N}x\pi)cos(\frac{2v+1}{2N}y\pi) \tag{2.3}$$

where $x, y, u, v = 0, 1, 2, ..., N-1$, and

$$c(x) = c(y) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } x = 0, y = 0 \\ 1 & \text{otherwise} \end{cases}$$

The inverse DCT (IDCT) transformation formula is given below:

$$f(u,v) = \frac{2}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} c(x)c(y)F(x,y)cos(\frac{2u+1}{2N}x\pi)cos(\frac{2v+1}{2N}y\pi) \tag{2.4}$$

The 2D DCT is used in JPEG image compression. During the encoding process, a JPEG image is split into YCbCr component instead of RGB component and the 2D DCT is applied to 8×8 blocks of YCbCr components of an image. A sample Y component values of an image and its corresponding DCT coefficient matrix is given in Table 2.6 and in Table 2.7 respectively.

In the 8×8 DCT coefficient matrix, the (0,0) element or the top-left element is called the DC or zero-frequency component, and the other components are called the AC component. From the Table 2.7, it is clear that the larger DCT coefficient values are concentrated in the upper left corner. The farther we get from the DC coefficient in the upper left corner, the smaller the values become. The DC coefficient is nearly three times as large as any of the AC coefficients.

Table 2.6: *Y component of YCbCr components of an image*

58	45	29	27	24	19	17	20
62	52	42	41	38	30	22	18
48	47	49	44	40	36	31	25
59	78	49	32	28	31	31	31
98	138	116	78	39	24	25	27
115	160	143	97	48	27	24	21
99	137	127	84	42	25	24	20
74	95	82	67	40	25	25	19

Table 2.7: *DCT coefficient matrix of a* (8×8) *image block*

-603	203	11	45	-30	-14	-14	-7
-108	-93	10	49	27	6	8	2
-42	-20	-6	16	17	9	3	3
56	69	7	-25	-10	-5	-2	-2
-33	-21	17	8	3	-4	-5	-3
-16	-14	8	2	-4	-2	1	1
0	-5	-6	-1	2	3	1	1
8	5	-6	-9	0	3	3	2

After calculating the DCT coefficients, the next step is to discard the coefficients that contribute the least to the image. The JPEG standard defines a mechanism, known as quantization, to do this. To quantize the DCT coefficients, they are actually divided by another value and rounded to the nearest integer. JPEG uses a 64 element 8×8 matrix to define quantum values for an image. It is called a quantization table . It is possible to use separate quantization tables for separate components of an image. The DCT coefficients are quantized by the corresponding quantum values in the quantization table. The standard quantization tables for Y, Cb, Cr components are given in Table 2.8 and Table 2.9 respectively.

Table 2.8: *Sample quantization table for Y component*

16	16	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	14	16	24	40	57	69	56
14	14	22	29	51	87	80	62
18	18	37	56	68	109	103	77
24	24	55	64	81	104	113	92
49	49	78	87	103	121	120	101
72	72	95	98	112	100	103	99

Table 2.9: *Sample quantization table for Cb and Cr components*

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

When Y component DCT coefficient matrix in Table 2.7 is quantized with the Table 2.8, the quantized DCT coefficient matrix in Table 2.10 is formed.

To produce the longest runs of zero values of the DCT coefficient and group them together, AC coefficients in a data unit are encoded using a zigzag path. Quantization values within quantization tables are stored in JPEG files using this zigzag ordering also. The zigzag ordering of the quantized AC coefficients is 18, -9, -3, -8, 1, -3, 1, -2, 4, -2, 4, 0, 3, -1, 0, 1, 1, 0, -1, -1, 0, 0, 0, -1,...then all are zeroes. After this, the quantized DCT coefficients are processed with any Run Length Encoding (RLE) or Huffman encoding, etc. before storing it as a JPEG file.

Table 2.10: *Quantized DCT coefficient matrix of Y component*

-38	18	1	-3	-1	0	0	0
-9	-8	1	3	1	0	0	0
-3	-2	0	1	0	0	0	0
4	4	0	-1	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

In DCT based steganography, the secret bits are actually embedded into the LSB of the DCT coefficients before creating the stego image. During the extraction phase, the DCT coefficients from the stego image are again formed, and the secret bits are extracted from the coefficients.

2.5 Discrete Wavelet Transform (DWT)

The wavelet transform converts a spatial domain image into a transform domain. The wavelet transform is done by the repeated filtering of the image coefficients on a row-by-row and a column-by-column basis. The main reason why wavelets are used in image steganography is that the wavelet transform fairly separates low-frequency and high-frequency information on a pixel-by-pixel basis. When a cover image is passed through a wavelet filter, the image is twisted with a wavelet low pass filter, granting smoother versions of the original input image or it will be twisted with a high pass filter, resulting in a final detailed band. On the other hand, final level decomposition for an image of low pass coefficients constitutes an approximation band.

In discrete wavelet transforms (DWT), the one-dimensional signal is divided into two parts; the high-frequency part and the low-frequency part. After that, the low-frequency part is split into two additional parts, and the analogous process will go on until reaching the desired level. The high-frequency part of the signal actually contains the signal's edge components. In each level of the DWT decomposition, an image is separated into four other parts which are referred to as the approximation image (LL), in addition to the horizontal (HL), the vertical (LH) and the diagonal (HH). In a DWT decomposition, an input signal must be the multiple of 2^n where n is the number of levels.

Haar wavelet is the simplest and most commonly used wavelet transform. It can be performed in two different ways; the horizontal way and the vertical way. Haar wavelet actually functions by scanning the pixels from left to right in a horizontal direction. Next, it will perform an

addition and subtraction operation on the neighboring pixels which multiplied by a scaling function for Haar wavelet is $1/\sqrt{2}$. At last, the final result of the addition on the left half and the addition on the right half will be stored. The process is repeated until it covers all the rows. The sum of the pixel is represented by a low-frequency, and the difference of the pixel is represented by a high-frequency. After completing the steps described, it is possible to scan the pixels from top to bottom in a vertical direction. In the end, the addition and subtraction operation will be multiplied by $1/\sqrt{2}$, and the result of the addition on the top and the subtraction on the bottom will be stored.

The two-dimensional DWT is a one-dimensional DWT for a two dimensional signal. It operates only on one dimension at one time, by analyzing the rows and columns of an image in a separable style. At first, a filter is applied to the image rows producing another two new images, where one image represents coarse row coefficients, and the other one represents detailed row coefficients. After that, an analysis filter is applied to the columns of each new image, producing four different images called the subbands. The rows and the columns are analyzed with a high pass filter (H). In the same way, rows and columns are analyzed with a low pass filter (L). For instance, if a subband image is produced by employing a high pass filter on the rows and a low pass filter on the columns, it is called the HL subband. Like this, 2D DWT actually generates four such subbands, LL, HL, LH, and HH.