An Enhancement of digital image steganography based on PVD and modulo operation using histogram

With the help of PDMO, APVD and PVD.

Major Techniques used:

# Embedding Process techniques:

## PDMO (Pixel Differencing Modulo Operation)

PDMO is a technique used mostly in steganography to embed or conceal a value within a picture. It is hidden in such a way that it is visually hard to determine whether the image contains any value. To accomplish this, we first compute the difference between the pixel values and then apply modulus, so that we can determine the value of the difference using certain rules, which we will cover later. This value is then used to embed the binary bits.

### PDMO Process

Assume that there are two pixels to which the PDMO approach should be applied. The following are the specifics of the techniques:

1. Calculate the difference in pixels . Additionally, if a negative difference occurs, it will be trivial to adjust its absolute value, as symmetry dictates that all conceivable values are equally relevant. .
2. Consider and to be the number of secret data bits to be implanted on the pixels and . Using the difference value , determine and as and bits, respectively, from Table I.

Table 1 :Range Table for This Method

|  |  |  |
| --- | --- | --- |
| Range ( |  |  |
| Capacity, |  |  |

1. Find the remainders and using following rules:
2. Convert and which is a part of binary secret message, to decimal say and respectively. Compute the difference values and as
3. Apply the specified criteria to the pixels andat the first level to obtain the consequent pixels and .
4. Calculate as new difference value.   
   If apply following,  
   and if apply following,
5. To resolve the condition of FOB, apply the following:
6. The modified pixels are and.

## APVD (Average Pixel Value Differencing)

Subsequently, two pixels (and) are required for computing the average of the pixels' values (), and a third pixel () is required for applying PVD. Thus, we will want three pixels in total. The following are the steps for applying the APVD:

1. The average () of two pixels (and) is calculated
2. Apply PVD to and . Let and are the altered pixels after applying PVD.
3. Calculate the difference value between and as
4. Now compute and using

## PVD (Pixel Value Differencing)

1. Assume and be the two consecutive pixels of a block.

Table 2 :Range Table for Wu & Tsai

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Range |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| Capacity, | 3 | 3 | 4 | 5 | 6 | 7 |

1. Say difference between and be as
2. Refer table 2 for to find range . Number of bits to be hidden in a block is computed as , where is the lower bound and is the upper bound for the range .
3. Let is the decimal value of bits of secret data. The new difference is obtained by .
4. Let be the difference between new and original difference values, i.e., .
5. The stego-pixels and can be obtained using

## Overflow and Underflow case resolution

In the case of overflow, obtain the largest pixel which exceeds 255 and compute the difference value using where signifies the larger value between and.

Now readjust the pixels and as

,

In case of underflow, obtain the smallest pixel which is less than 0 and compute the difference value using, where signifies the smallest value between and .

Now readjust the pixel values and using,

# Extraction Techniques

The extraction process is essentially the reversal of the embedding process which approaches to find out the bit embedded in the proceeding steps. The following methods have been applied to retrieve the embedded bits.

## Reverse PDMO (Pixel Differencing Modulo Operation)

PDMO is a technique to embed secret message in the given pixels which is mentioned in the text by Sahu & Swain. Let and be the stego-pixels of a block as shown in the following fig.

|  |  |  |
| --- | --- | --- |
|  |  |  |

Calculate the difference value . Now obtain the remainder values and using following equations:

If then represent and in 3 binary bits, otherwise, represent and in 4 binary bits. These are extracted binary bits from stego-pixels and .

## Reverse PVD (Pixel Value Differencing)

Let the stego-pixels and , find the difference as . The value falls in one of the ranges of range of the table 2. Let be the difference between and its corresponding lower bound , i.e., . Represent to bits, which is extracted bits.

## Reverse APVD (Average Pixel Value Differencing)

It basically works with some other algorithms which are either used to modify the pixel values or to embed bit of message value into pixels as it may be. Initially Let and and be the stego-pixels of a block as shown in the following fig.

|  |  |  |
| --- | --- | --- |
|  |  |  |

The average stego-pixel can be found using . Next step is to apply reverse PVD to find the bits embedded in the and .

# Proposed Algorithm

## Embedding algorithm

Algorithm follows as:

1. Select the pixels as , , , and. Secret Message .
2. Argument list:  
    input pixels for Pixel Differencing Modulo Operation is secret message in binary form. is starting point of the secret message.   
   Returning values:  
   , modified pixels which is altered by PDMO function. is end point of secret message which already embedded using PDMO.
3. Average pixels,
4. Argument’s list:   
    is input pixel for Average Pixel Value Differencing. is starting point for secret message to embed.  
   Return’s list:  
    altered pixels by APVD is end point of secret message.
5. Argument’s List:  
    is input pixels for Pixel Value Differencing  
    is secret message.  
    is starting point for secret message.  
   Return’s List:  
    is altered pixels by PVD, is end point of secret message.
6. *Stego-pixels:* , , ,

## Extraction Algorithm

1. The Stego-pixels: , , ,
2. is argument for Reverse PDMO.  
    first part extracted secret message
3. is argument for Reverse APVD.  
    first part extracted secret message
4. is argument for Reverse PVD.  
    first part extracted secret message
5. Concatenate the 3 parts of secret message
6. Extraction is completed.