Shuffling of similar pixels and concealing image with equivalent segment support

[1]Narisimhamurthy.Yenda [2]R.Srinivas

[1](Final Year M.Tech Student,Dept. of CSE,Aditya Institute of Technology and Management(AITAM),
Tekkali,Srikakulam,Andhra Pradesh, email: narisimhamurthy.yenda@gmail.com)
[2](Associate Professor, Dept. of CSE,Aditya Institute of Technology and Management(AITAM),
Tekkali,Srikakulam,Andhra Pradesh ,email: srinuk2008@yahoo.com)

Abstract:

Always the encoding or putting secret keys in images are the big challenges. So the image processing is the biggest open challenge for IEEE persons. Here in this paper we are doing research on how best an image can be concealed with secret key for low level pixel pairs and readjusting the pixel pairs to overcome the disturbance for preceding the pixel pairs. Fast pixel pair match technique is used to frame the all available and possible pixel pairs. And make a random key for the canceling to replace the pixel pairs. So here segmentation is framed in percentage wise to know all the pixel relative duplications. So after the segmentation the shuffling of the pixel pairs will be easy and handy for further concealing the data digits, this leads to proper adjustment of pixel pairs and also fast concealing of the data digit. *(Index terms: pixel pair matching, concealing, segmentation)*

PROPOSED WORK

The new approach the FPPMT (fast pixel pair match technique) is proposed to prepare the all available pairs/duplicates and maintain in LOG. Once the pixel pairs manufactured when we are concealing the data digit (random digit which is unique for all pixel pair categories) into all available pixel pairs the preceding pixel gets disturbed with the concealed data digit. This rises when we shuffle the pixel pairs for readjustment. So before shuffling we propose a new technique to put segmentation for pixel pairs. These segmentations are unique with each other to put the pixel pairs without disturbing the preceding pixels by taking the percentage match category. So after we have segmentation the data digit will be concealed. So this reduces for not only for making the pixel pairs but also to get the image to the normal form easily. This technique is very handy for further image compression after the data digit concealing and also for deflations of the image.

RELATED WORK

In this project we are using an innovative approach of merging two images in order to encrypt the secret information in the images this is purely to hide the secrecy in data with reference to the forgery problems. Though there are many works related to image security but this method will find a new way of approach to hide the data in images and then again decodes them in order to save the original data which can be utilized in future.

FPPMT:

Normally starting with the first pixel by ending with last pixel, our work first scatters the pixel pair matching to segregates them into segmentation (@). Starting with the first pixel preceding pixel will be framed in the temporary storage buffer to prepare next temporary storage buffer (if 3 pixels with naming ***p1, p2, p3ie*** 2 pixel pairs ***(p1,p2) & (p2,p3))***for comparison to retrieve the similarity pixel pairs based on the threshold given in our work. Once these pairs are matched with our proposed condition for the possible pixel pair matching vector$\sum P_{np}$ will be framed by our system. This vector contains the possible pixel pairs with our threshold condition. But main pixel will be in the$\sum P$.

The transactional information will be in the log information stack just without losing any micro information of the process in the transactions. The main usage of this log is to maintain the consistency of the image without distortions after retain to normal form. This is per image and will be deleted for next level of process.

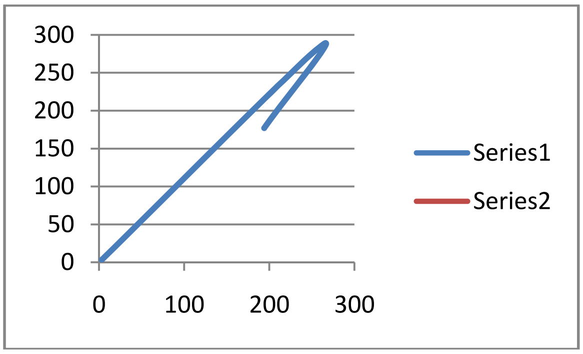
1) $\sum_{o}^{n} \mathbf{p} \rightarrow \{\text{Total no of pixels in vector}\}$
2) $\sum_{o}^{i} p_{\mathrm{np}}$→{ Total no of possible pixel pairs in a vector}
3) r_{k}←random data digit
4) L_{m}←log information stack
5) $S_{\mathrm{G}}(\mathrm{n,t})$ ← segmentation

$\left\{ \begin{array}{l} n \leftarrow \text{offset of segmentation} \\ t \leftarrow \text{segmentation range} \end{array} \right\}$

6) λ ← threshold

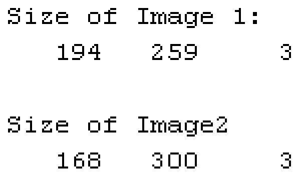
7) $\sum_o^n t_p$←total no of pixel pair vector

8) m←mark



Graph:

In the graph we can visualize the plotted points by using the values that are being displayed in the screen shot. These values are nothing but the size of an image.



```
Size of Image 1:
   194   259     3

Size of Image2
   168   300     3
```

Initialization state

step1:

l←0//(temporary location where each and every pixel pair will be saved segmentation)

$\sum P_{np}$←0

t←0//location where each pixel temporary stores

Count = 0

Step 2:

For each $t_i \leftarrow \sum_o^n P//$outer loop

$t_{i+1} \leftarrow \sum_0^n P$//outer loop

(t_i,t_{i+1})

Inner loop:

For each $L_{oi} \leftarrow \sum_0^n p(i+2)$

$l_{oj} \leftarrow \sum_0^n p\ (i+3)$

(l_{oe},l_{oj})

If$(t_i,t_{i+1})==(l_{oe},l_{oj})$

P $\leftarrow(t_i,t_{i+1})$

```
++count
Else $(t_{i,}t_{i+1})\neq(l_{oi},l_{oj})$
$L_{oi}\leftarrow i+n$
$l_{oj}\leftarrow i+n+1$
End of out loop.
```

Starting the in the loop with respect to all the pixels in the range this part will be explanatory for pixel pair matching criteria after log information. For each and every pixel in the ***P*** vector pixels will be made of pixel pairs and all the pixel pairs which are in the possible pixel pairs ***P_{np}*** will be compared to get the comparison results for the next level segmentation. The condition of the comparison is as follows with this condition ***$[(t_{i,}t_{i+1})==(l_{oe}, l_{oj})]$*** . Basically left parameter will be the outer loop or main pixel loop and right parameter is from inner (ie possible pixel pair matching vector). If this matches this will be put up in the temporary vector for next levels ie segmentations and to have range stack.

(@) segmentation:

```
Step 1:
          →Vector
 For each i← $\sum_{np}^{i} P$
If Pn∈ $S_o$
$S_o\leftarrow P_n$
Else if $P_n \in S_1$
$S_1\leftarrow P_n$
Else if $P_n \in S_2$
$S_2\leftarrow P_n$
Else
$S_3\leftarrow P_n$
```

Pixel pair range:

```
Step 1:
N←0
N-number of effective pixel pair distortion where adjacent the current pixel pair
For each pair in $p_p$
Loop
$t\leftarrow P_{np}$
Range=shuffle (Pp, R0)
If $0\leq range\leq 25$
$S_0$=t
{
Else if
$25<range\leq 50$
S1=t
```

```
Else if
$50<rang\leq 75$
$S_2$=t
Else
$S_3$=t
```

Segmentation	Range
S_0	0-25%
S_1	26-50%
S_2	51-75%
S_3	76-100%

Segmentation table (***50***)

We assume our segmentation table ranges in the above table*50* to segregate the pixel ranges in the form of increasing wayS0, *S1, S2, S3* are the four ranges in our proposed work to have these pixels to have concrete comparison with the log in formations after they come back to normal form after decryption.

Detailed description for FPPMT:

Image encryption and decryption

Image formats are .bmp and .tiff in the proposed paper.

Concealing key

Byte key [ ] = {0x8E,0x12,0x39,0x9C,0x07,0x72,0x6F,0x5A}

```
    ECIPHARINSTANCE=”DES/CBC/PKCS5 Padding”
    DCIPHARINSTANCE=”DES/CBC/PKCS5 Padding”
    Cipher encryption initialization mode←INIT (mode,key, paver format)
    Cipher decryption initialization mode←INIT (mode,key, paver format)
```

Encryption

```
    Initialization
    $\sum_{0}^{1024} buf \leftarrow 0$
    num←0
    Count←0
    ENCR (In, Out)
    Do Loop Start
    buf←read(in)
    num←buf
    Out(WRITE(num))
    End loop
```

Decryption

```
    Initialization
    $\sum_{0}^{1024} buf \leftarrow 0$
    num←0
    count←0
```

```
ENCR (In, Out)
Do Loop Start
buf←read(in)
num←buf
Out (WRITE (num))
End loop
```

Encryption and decryption: normally encoding and decoding will be used in where there is a need of encode our represented binary data that’s needs to be preserved or for communication using local or remote media networks. This is done to the data(encrypted) to restrict for the further modifications during above said operations. In this encryption technique originally system will take slices of sequential bits. This will take ***8*3*** multiplication bits individually. And then these will be sliced into 6 bits length slices. The asci conversion of 3-byte ***8*3***multiplication bits is repeated until the whole sequence of main data bytes are encoded. If number of bytes are giving reminder for the case of 3 division then extra bytes with 0 values and perform the regular operation. If only one single byte the most significant bit will be padded with some mathematical symbol. The ‘==’ symbol will be appended for 2byte data and ‘=’ will be appended to 1byte data.

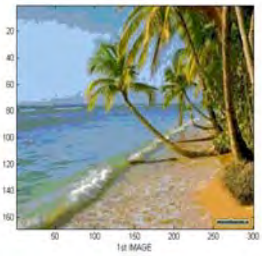
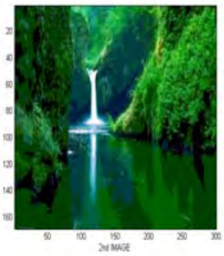
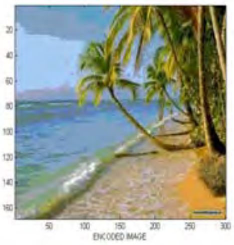
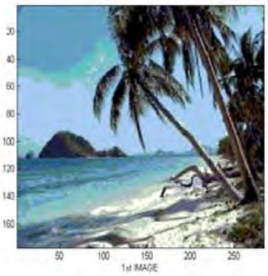
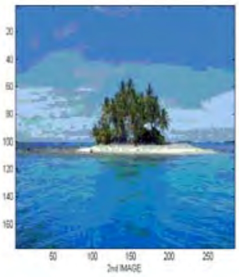
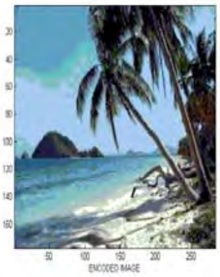
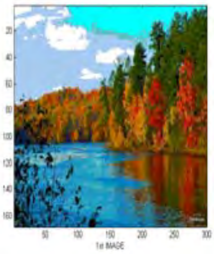
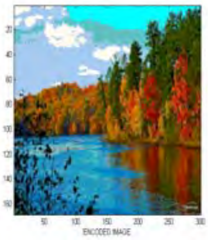
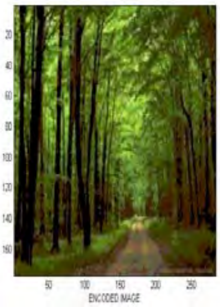
CONCLUSION

In our paper concealing the key in the given image is taking an unique approach. In this approach double or twin pixels(matched pixels) will be concealed with the secret key which is unique in our technique. This secret key is also random for every generation. We have segmentation to have duplicated or twin in categorical blocks in percentage wise. Now for transmission we use encryption 8*3 tequnique to encode and decode. So the image is now is used for internal or external transmission over the net works. This is totally new approach which is not there in any of the previous work. So by using this work we can have fair concealing encryption for the further processing of the image.

FUTURE WORK

In the future approaches we may use the same techniques to be applied on images even though they are on different networks. As this ensures that image resolution will not affect and remains same with same range of pixels and mapping. This gives a trusted security for the encrypted images.

EXPERIMENTAL RESULTS

Difference in time	Image 1	Image 2	Sizes:	Output
0.0001			Image1:230kb Image2:230kb	
0.0004			Image1:230kb Image2:230kb	
0.0005			Image1:230kb Image2:230kb	
0.0004			Image1:230kb Image:230kb	

REFERENCES:

[1] Ballard (1982), Computer Vision --TA1632.B34

[2] Baxes, Gregory A (1994) DIGITAL IMAGE PROCESSING, Wiley *IT* (includes disk with PC software)

[3] Castleman, Kenneth (1979), Digital image Processing Prentice-Hall, --TA1632.C37-- (recommended by UTHSCSA Image Tool)

[4] Foley, J.D., VanDam, A. (1982) FUNDAMENTALS OF INTERACTIVE COMPUTER GRAPHICS, Reading Massachusetts, Addison-Wesley, chapter 17.

[5] Gonzalez,R.C. (1977), DIGITAL IMAGE PROCESSING, Reading Massachusetts: Addison_Wesley.

[6] Pavlidis, T. (1982) ALGORITHMS FOR GRAPHICS AND IMAGE PROCESSING, Computer Sci. Press,Rkvl., MD.

[7] Pratt, W. K. (1978), DIGITAL IMAGE PROCESSING. New York: Wiley-Interscience. TA1632.P7

[8] Rosenfeld, A. and Kak,A. C. (1982), DIGITAL PICTURE PROCESSING, Second Edition, Vol. 1,2. NY:Academic Press, TA1630.R67

[9] Russ, J.C. (1986), PRACTICAL STEREOLOGY. New York: Plenum Press
[10] Russ, John C.(1995), THE IMAGE PROCESSING HANDBOOK, 2nd ed., CRC Press. *IT*

BIBILIOGRAPHY:

Narisimhamurthy Yenda received MCA from GMRIT, Rajam, Srikakulam, Andhra Pradesh. Currently he is pursuing M.Tech in Adithya Institute of Technology and Management, Tekkali,Srikakulam, Andhra Pradesh.

R.Srinivas is an Associate Professor in Adithya Institute of Technology and Masnagement, Tekkali, Andhra Pradesh.