HIGH CAPACITY REVERSIBLE DATA HIDING USING OVERLAPPING DIFFERENCE EXPANSION

E. Chrysochos, E. E. Varsaki, V. Fotopoulos, and A. N. Skodras

Digital Systems & Media Computing Laboratory,
School of Science and Technology, Hellenic Open University,
13-15 Tsamadou st., GR-26222, Patras, Greece
phone: + (30) 2610 367535, fax: + (30) 2610367520, email: {e.chrysochos,e.varsaki,vfotop1,skodras}@eap.gr
web: dsmc.eap.gr

ABSTRACT

Difference expansion (DE) has been widely used for reversible data hiding. In this work, a new DE based scheme is presented that uses consecutive, overlapping pairs, instead of the non-overlapping pairs or triads used by traditional DE derivatives. The scheme is superior to the existing approaches, both in capacity and PSNR terms. By applying multiple runs of the embedding process, a significant capacity gain is obtained at the expense of lower quality.

1. INTRODUCTION

Data hiding deals with the ability of embedding data into a digital cover with the minimum amount of perceivable degradation. That is, the embedded data are invisible or inaudible to human observer. The invisible data (message) and the carrier medium (cover or host), could be text, image, audio or video. This work focuses on image covers. Reversible, lossless or invertible data hiding embeds data into a digital cover in a way that the message can be error-free extracted from the cover-medium, while the carrier is restored to its original form. The reversibility property in data hiding methods, could find application in the fields of copyright protection, secure medical image data systems, image authentication, tamper proofing etc. [1].

Such data hiding techniques can be implemented both in spatial and frequency domain [2]. In spatial domain Ni et al. [3] proposed a technique based on the modification of the image histogram. Chrysochos et al. [4] presented a scheme, robust to geometrical attacks, based on histogram modification. Lee et al worked on the histogram modification of difference image in [5]. Tian developed a novel data hiding technique using Difference Expansion (DE) [6]. The DE method's basics will be described in section 2.1. Shen et al proposed a new scheme based on the Difference Expansion, called 3C2B [7]. This scheme provided better capacity in comparison to Tian’s algorithm at the expense of lower PSNR values. Alattar modified Tian’s DE in triplets [8]. A step further is made by Lin et al. by avoiding the location map on DE scheme [9]. Other techniques in the frequency domain appear in [10] where the histogram of the integer wavelet transform is modified.

The proposed scheme performs in spatial domain and is based on DE principle. Traditional DE methods apply DE on non overlapping pixel pairs [6] or triads [7]. On the contrary, the proposed method uses successive overlapping pairs of pixels resulting in a significant capacity gain. Difference expansion theoretically can give up to 0.5bpp, while 3C2B scheme approaches 0.67 bpp. The proposed method may perform up to 1bpp, outperforming the aforementioned techniques, in terms of capacity versus PSNR values, while preserving reversibility.

The rest of the paper is organised as follows. In section 2 the embedding procedure is described. The extracting procedure is explained in section 3 and capacity issues are discussed in section 4. Experimental results are presented in section 5, while conclusions are drawn in section 6.

2. EMBEDDING PROCEDURE

Let an 8 bit greyscale image of MxN pixels and pixel values \(x(m,n)\), where \(m \in [0,M), n \in [0,N)\). The proposed Overlapping Difference Expansion extends the Difference Expansion method of Tian [6]. Tian uses non-overlapping pairs of pixels, while the proposed scheme uses successive overlapping pairs. We can compute the Overlapping Difference Expansion of a triad of adjacent pixels \(x_1, x_2\) and \(x_3\). First we compute the DE integer transform of the first two pixels by defining their integer average \(a_1\) and their difference \(d_1\) as:

\[
\begin{align*}
a_1 &= \left\lfloor \frac{x_1 + x_2}{2} \right\rfloor \\
d_1 &= x_1 - x_2
\end{align*}
\]  

where the symbol \(\lfloor \cdot \rfloor\) represents the floor function. The expanded difference \(d_1'\) is given by

\[
d_1' = 2d_1 + w_1
\]  

where \(w_1 \in \{0,1\}\) is the first watermark bit. The new values of the first pair \(x_1'\) and \(x_2'\) are given by

\[
x_1' = a_1 + \left\lfloor \frac{d_1' + 1}{2} \right\rfloor, \quad x_2' = a_1 - \left\lfloor \frac{d_1'}{2} \right\rfloor
\]  

Then we compute the DE integer transform of the new \(x_1'\) with the \(x_3\) pixel by defining their integer average \(a_2\) and \(d_2\) as:
The expanded difference $d'_i$ of the new $x'_i$ with the $x_i$ pixel is given by

$$a_i = \frac{x'_i + x_i}{2} \quad \text{and} \quad d'_i = x'_i - x_i \quad (4)$$

The new values of the second pair $x'_i$ and $x'_i$ are given by

$$x'_i = a_i + \left\lfloor \frac{d'_i + 1}{2} \right\rfloor, \quad x'_i = a_i - \left\lfloor \frac{d'_i}{2} \right\rfloor \quad (6)$$

There is a possibility that in some cases the transformed values may exceed the highest value of 255 or the lowest of 0, producing overflows or underflows, respectively. In order to maintain the reversibility of the DE transform the following conditions must be satisfied:

$$0 \leq a_i + \left\lfloor \frac{d'_i + 1}{2} \right\rfloor \leq 255 \quad \text{and} \quad 0 \leq a_i - \left\lfloor \frac{d'_i}{2} \right\rfloor \leq 255 \quad (7)$$

where $i = 1, \ldots, (M \times N)$.

As mentioned before, difference $d$ is expanded to $d'_i$ according to equation (2). Therefore a difference value $d$ is expandable under the integer average $a$ if

$$2d + w_i \leq \min(2(255 - a), 2a + 1) \quad (8)$$

By this watermark bit $w$ can be reversibly embedded.

As stated above, pairs of pixels that produce overflow or underflow values are excluded from the embedding procedure. Another case of embedding exclusion is the one where the DE input pixel values difference exceeds a predefined threshold. Given a threshold of value $T$, the new set of pixels can not differ more than $2T+1$. For example for $T=10$ the pixel intensities may differ up to 10, and the resulting intensities’ difference can not exceed 21. This alteration may or may not be perceivable, depending on the image content. For noisy images a higher threshold may be selected with no significant degradation of the resulting image, while for smooth images, lower thresholds are more appropriate. Another aspect of the threshold selection has to do with the resulting capacity. A higher threshold results in higher capacity (but lower PSNR) whereas lower threshold results in lower capacity (but higher PSNR). Therefore the threshold can be fine-tuned according to the acceptable perceptual degradation of the image and the capacity requirements.

During the embedding procedure a binary valued location map is formed. This map has the same dimensions with the original image. A value of 0 is assigned for each acceptable pair of pixels, whereas a value of 1 is assigned otherwise. The last case relates to underflows, overflows and threshold constrains. The resulting map is necessary for the extraction procedure as it locates the positions, where the successive embedding chain is broken. Thus the map is compressed in a lossless manner, using either JBIG2 or bi-level TIFF compression format, and is added to the message to be embedded.

As the above procedure is fully reversible, it could be repeated more than once in order to increase capacity. Besides the embedding procedure could be extended to triads or even to N-ads, but this is a topic for future work.

3. EXTRACTING PROCEDURE

To extract the message and restore the pixel values, the average and difference computations are performed on the pixel pairs of the stego-image, recursively. An embedded bit is extracted from each pair by checking the LSB bit of their difference. If the difference is odd then the bit extracted is ‘1’. If the difference is even then the extracted bit is ‘0’. Thus the following formula is used:

$$w_i = d'_i \mod 2, \quad (9)$$

where $w_i$ is the recovered watermark bit and $d'_i$ the difference between pixels of a pair. In order to restore the pixel intensities of a triplet we first restore the last expanded pixels $x'_2$ and $x'_3$ to their previous state. Their average $a'_i$ is calculated by (10) and their restored values $x'_2$, $x'_3$ by (11).

$$a'_i = \frac{x'_2 + x'_3}{2} \quad (10)$$

$$x'_2 = a'_i + \left\lfloor \frac{d'_i + 1}{2} \right\rfloor, \quad x'_3 = a'_i - \left\lfloor \frac{d'_i}{2} \right\rfloor \quad (11)$$

Then the pixels $x_1$ and $x_2$ are restored by repeating the same step described by equations (10) and (11).

Extraction pairs are selected in reverse order of embedding. As the whole embedding procedure may be iterative, the same number of extraction steps has to be performed in order to acquire back the original image and the message. The extraction procedure consists of three steps.

- The location map is first extracted and decompressed. This map informs the decoder about the pixel pairs that carry (or not) information to be recovered.
- Next, the message from the pixel pairs is extracted according to the location map. The pairs are used in reverse order of encoding. When all the embedded bits are collected the extracted message is formed.
- The original image is recovered by using the location map information and equations (10), and (11).

If the embedding is iterative, the extraction procedure recovers the input image of the latest run and part of the message. It will take the full number of iterations in order to get the original image and the full message back.

4. CAPACITY ISSUES

The proposed algorithm can embed theoretically up to one bit per pixel. This is the case were all the pixels are used by the algorithm for embedding; so one bit is embedded per pixel. For an image of size $M \times N$, the total capacity $C_t$ is given by

$$C_t = M \cdot N - P_u \quad (12)$$

where $P_u$ is the number of pixel pairs that were excluded due to underflows, overflows and threshold constrains. Neverthe-
less in order to compute the pure capacity \( C \) for a given image the size of the compressed location map (\( C_m \)) should be subtracted from \( C_t \), i.e.

\[
C = C_t - C_m,
\]

(13)

For the 512x512 Lena image, the location map when compressed in JBIG2 format does not exceed 15KB. This corresponds to nearly 50% of the available embedding capacity. The capacity limit of 1bpp can be surpassed by implementing iterative embedding (multipass), that is, by applying each time the embedding algorithm to the image resulting from the previous run. Since the scheme is fully reversible, each time the extraction algorithm is applied on the image, the later reverts to the stage of the previous run. The procedure is repeated as many times as necessary, in order to revert back to the original image and extract the full message. Each run produces a theoretical capacity of up to 1bpp. By performing \( L \) iterations, the theoretical capacity can go up to \( L \) bpp.

5. EXPERIMENTAL RESULTS

The proposed data hiding scheme has been implemented in Matlab (v7.0). The watermark used for embedding was a random sequence of ‘0’ and ‘1’. The location map was compressed by lossless JBIG2 format and was added to the embedded message. Table I presents the capacity, the quality measurements (PSNR) and the perceptual transparency of the watermark (WPSNR) of different images for \( T=10 \). There is no visual degradation of the watermarked images, as high PSNR values are achieved.

<table>
<thead>
<tr>
<th>Image</th>
<th>( C_t ) [bpp]</th>
<th>PSNR [dB]</th>
<th>WPSNR [dB]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cameraman (256x256)</td>
<td>0.7582</td>
<td>40.80</td>
<td>62.28</td>
</tr>
<tr>
<td>Jelly beans (256x256)</td>
<td>0.8989</td>
<td>46.45</td>
<td>61.90</td>
</tr>
<tr>
<td>Girl (256x256)</td>
<td>0.8767</td>
<td>41.36</td>
<td>59.74</td>
</tr>
<tr>
<td>Moon surface (256x256)</td>
<td>0.7035</td>
<td>39.80</td>
<td>60.27</td>
</tr>
<tr>
<td>Lena (512x512)</td>
<td>0.8347</td>
<td>40.37</td>
<td>60.66</td>
</tr>
<tr>
<td>Clock (256x256)</td>
<td>0.8670</td>
<td>42.36</td>
<td>61.87</td>
</tr>
<tr>
<td>Boat (512x512)</td>
<td>0.7064</td>
<td>39.65</td>
<td>60.80</td>
</tr>
<tr>
<td>Elaine (512x512)</td>
<td>0.7413</td>
<td>40.02</td>
<td>60.67</td>
</tr>
<tr>
<td>Lighthouse (512x512)</td>
<td>0.6821</td>
<td>41.34</td>
<td>61.32</td>
</tr>
<tr>
<td>Straw (512x512)</td>
<td>0.3682</td>
<td>42.99</td>
<td>62.04</td>
</tr>
<tr>
<td>Baboon (512x512)</td>
<td>0.5215</td>
<td>41.44</td>
<td>60.90</td>
</tr>
<tr>
<td>Peppers (512x512)</td>
<td>0.8081</td>
<td>39.10</td>
<td>61.32</td>
</tr>
<tr>
<td>F16 (512x512)</td>
<td>0.8760</td>
<td>42.00</td>
<td>60.97</td>
</tr>
<tr>
<td>Pentagon (1024x1024)</td>
<td>0.7121</td>
<td>39.80</td>
<td>60.88</td>
</tr>
</tbody>
</table>

The embedding algorithm was applied on Lena for different threshold values (\( T \)), as shown in Fig. 1. As expected, capacity increases while PSNR decreases. For \( T=255 \) the theoretical bound of 1bpp is approximated. Above a threshold value of \( T=60 \), there is no practical capacity gain. A good trade off between capacity rate and PSNR value is achieved close to threshold value of 10. Even with \( T=255 \) and capacity of 0.9994, watermarked ‘Lena’ has a quite acceptable quality, as shown in Fig. 2. In this case PSNR=35.19 dB, while the image seems like being high-pass filtered. Distortion becomes more visible in high frequency regions like the hat’s feather.

In such schemes, there is always a trade off between capacity and quality. If capacity is the main goal, smoother images are better hosts. This is because comparing to a noisy image, for the same threshold value, in the smooth one, less pairs are excluded. As shown in Fig. 3, for the same threshold, the smooth Lena curve is always to the right (higher capacity) of the noisy baboon. At the same time, it is obvious that for the same, small thresholds, baboon’s PSNR is above that of Lena’s (because few pairs are changed). As the threshold increases, since more pairs are used in baboon, with largest differences produced due to DE, the PSNR gap is closing; at T=13 both images have the same PSNR but from this point on, the gap starts widening again but this time in the opposite direction. So if quality is the main concern and reduced capacity is not a problem, noisy images with low thresholds are good candidates. For a good trade off, a smooth image with a threshold in the range 10-13 is an excellent selection.

As mentioned earlier, higher capacity can be achieved by applying iterative embedding. For each iteration, the ex-
tra capacity gain is reduced and PSNR decreases, as expected. This is shown in Table II.

The proposed algorithm for one iteration has a theoretical upper bound of capacity reaching 1bpp, while [6] and [7] have 0.5 bpp and 0.667bpp, respectively. Furthermore, the experimental results of [6] and [7] are outperformed, by higher capacities and PSNR values. In the context of iterative embedding, the presented scheme still outperforms by far [6] and [7], reaching higher overall capacity, as shown in Fig. 4.

![Fig.3: PSNR vs Capacity for smooth and textured images with respect to threshold value \( T \)](image)

### Table II: Capacity and PSNR with respect to iteration for Lena image

<table>
<thead>
<tr>
<th>Iterations</th>
<th>Bit Rate per Iteration [bpp]</th>
<th>Overall Bit Rate [bpp]</th>
<th>PSNR [dB]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1(^{st})</td>
<td>0.8347</td>
<td>0.8347</td>
<td>40.37</td>
</tr>
<tr>
<td>2(^{nd})</td>
<td>0.5471</td>
<td>1.3818</td>
<td>35.82</td>
</tr>
<tr>
<td>3(^{rd})</td>
<td>0.2915</td>
<td>1.6733</td>
<td>34.31</td>
</tr>
<tr>
<td>4(^{th})</td>
<td>0.1697</td>
<td>1.8430</td>
<td>33.58</td>
</tr>
<tr>
<td>5(^{th})</td>
<td>0.1008</td>
<td>1.9438</td>
<td>33.19</td>
</tr>
</tbody>
</table>

![Fig.4: Overall capacity after iterative embedding in Lena image](image)

### 6. CONCLUSIONS

In this work an improved reversible data hiding scheme has been presented. The method is based on the well known difference expansion family of techniques. Its novelty involves the usage of consecutive, overlapping pairs, instead of the traditional non-overlapping pairs or triads. According to the experimental results, the proposed scheme outperforms its other DE based counterparts, both in terms of capacity and PSNR. Capacities higher than 1bpp can be achieved by multiple embedding runs.

### ACKNOWLEDGEMENTS

This work was funded by the European Union - European Social Fund (75%), the Greek Government - Ministry of Development - General Secretariat of Research and Technology (25%) and the Private Sector in the frames of the European Competitiveness Programme (Third Community Support Framework - Measure 8.3 - programme ΙΕΝΕΑ - contract no.03ΕΔ832).

### 7. REFERENCES


