Optimal look-up table-based data hiding

X. Wang  X.-P. Zhang

Department of Electrical & Computer Engineering, Ryerson University, 350 Victoria Street, Toronto, ON, Canada M5B 2K3
E-mail: xzhang@ee.ryerson.ca

Abstract: In this study, the authors present a novel data hiding scheme using the minimum distortion look-up table (LUT) embedding that achieves good distortion-robustness performance. LUT-based data hiding is a simple and efficient way to embed information into multimedia content for various applications, such as transaction tracking and database annotation. The authors find it possible to optimally reduce the data hiding-introduced distortion by designing the LUT according to the distribution of the host at a given robustness level. The authors first analyse the distortion introduced by LUT embedding and formulate its relationship with run constraints of LUT to construct an optimal coding problem. Subsequently, a Viterbi algorithm is presented to find the minimum distortion LUT. Then a new practical data hiding scheme using the optimal LUT is applied in the wavelet domain. Theoretical analysis and numerical results show that the new LUT design achieves not only less distortion but also more robustness than the traditional LUT-based data embedding schemes under common attacks such as Gaussian noise and JPEG compression.

1 Introduction

Data hiding techniques have been widely used in multimedia security applications such as copyright protection, authentication and transaction tracking. Many schemes have been proposed to fulfill the design requirements of various kinds of applications [1–3]. Robustness to signal processing operations, information payload and fidelity (or embedding introduced distortion) compose the three most important conflicting goals of a data hiding system. Achieving one property in most cases means sacrificing the others. Depending on specific applications, the desired data hiding methods need to achieve the trade-off among these three requirements.

In recent years, a lot of practical data hiding systems have been designed for image, video and audio and could be classified according to applications, working domain and embedding types. First, based on different robustness requirements of various applications, they can be classified into robust, fragile and semi-fragile data hiding. The copyright protection needs the scheme to be robust enough to survive malicious and non-malicious attacks. Recent robust watermarking research also focuses on different application aspects. In [4, 5], the lossless or reversible data hiding methods are discussed with application to medical recording and law enforcement. In [6], a spread spectrum watermarking scheme combined with a new perception model is presented to focus on imperceptibility requirement. On the other hand, fragile and semi-fragile techniques are fragile to some extent for the tampering detection applications. Second, different image domains including spatial domain and transform domain are available for hiding information. The data hiding in the transform domain [7, 8] can often achieve better perceptual transparency and robustness. Third, we could classify the data hiding system as non-informed and informed data embedding according to whether the information of the host is considered during embedding process. The additive spread spectrum algorithm [7, 9] belongs to the first category where the embedding process is independent of the host content. For the informed data hiding, the properties of the host are considered to force a relationship between the host signal and the information to be embedded [10, 11]. The idea is inspired by Costa’s dirty paper theory [12]. Quantisation-based methods [13–15] are in this category. The most favourable advantage of these methods is host interference rejecting. In quantisation-based methods, the information is embedded into the host by choosing information associated with quantisers to quantise the host data.

Look-up table (LUT) embedding is a simple and efficient quantisation-based scheme. The most popular LUT method is odd–even embedding or dither modulation [13]. It is a special case of scalar quantisation-indexed modulation (QIM), which is widely known in watermarking community. In [13], the distortion compensation QIM offers better performance over QIM, but the statistics of noise needs to be known in advance. In this paper, we consider the case with unknown noise statistics. Note that the distortion compensation will improve the performance of our scheme the same as in the QIM case when the noise statistics is known. LUT-based data hiding schemes have the following two main advantages: (i) the LUT is generally easy to implement and computationally efficient, and (ii) by constraining the quantisation points in a finite set in LUT rather than an infinite set (real-valued set) in a generic QIM, we show in this paper that we can better control the robustness distortion of the embedded data. The

The pixel-domain LUT embedding scheme is proposed in [15], ones in LUT. The run of the odd–even method is 1. A
is defined as the maximum number of consecutive zeros or
in this paper.

One of the most important properties of LUT is the run that is
defined as the maximum number of consecutive zeros or
ones in LUT. The run of the odd–even method is 1. A
pixel-domain LUT embedding scheme is proposed in [15],
where the LUT is associated with a cryptographic key to
provide security and it is a n-run LUT, that is, the
maximum allowable run of the LUT is n. Wu [16] indicated
that n-run LUT embedding generally introduces larger
distortion than the traditional odd–even embedding with the
same quantisation step size but provides more
robustness, that is, the bit error rate (BER) can be
considerably smaller. This conclusion is based on the
assumption that the host data follow a uniform distribution.
When the host data follow other distributions such as
Gaussian, it is possible to design LUT with less distortion
while maintaining the run length which is an indicator of
robustness.

In our previous work [17], we show that with the
knowledge the host statistics, the LUT can be designed to
achieve less distortion than existing schemes given a
robustness constraint defined by the run length. A reduced
distortion 2-run LUT is developed to achieve good
robustness and distortion trade-off. However, the solution is
limited to run of 2 and the distortion is reduced compared
to other methods but not minimised and the method cannot
be applied to arbitrary run length.

In this paper, a new generic optimal LUT embedding
method that minimises distortion for arbitrary run length of
LUT is presented. The LUT is generated with knowledge of
the information to be embedded. From the analysis of the
mean squared distortion introduced by n-run LUT, we show
that the distortion can be greatly reduced by designing
the LUT according to the distribution of the host data and
the data to be embedded. We further formulate the
minimisation of the LUT distortion as a dynamic
programming problem. Unlike the complex algorithm in
[17], a new practical minimum distortion n-run LUT design
method is presented based on a Viterbi algorithm (VA).
Experimental results show that at the same watermark-to
noise ratio (WNR), the BER for minimum distortion
n-run LUT embedding can be smaller than other LUT methods
including the odd–even LUT embedding.

The rest of the paper is organised as follows. Section 2
gives a brief introduction of data hiding and LUT-based
embedding. In Section 3, we analyse the distortion
introduced by data embedding in the LUT scheme. The
design of optimal (minimum) distortion LUT algorithm by
VA is proposed in Section 4. The optimal LUT is applied
to the wavelet domain in Section 5. Experimental results
with visual effects are given to demonstrate the advantage
of the new LUT embedding scheme over the existing
schemes in Section 6. Section 7 concludes the paper.

The new method is a general scheme that can be used for any
type of multimedia content since it takes advantage of the
distribution information of host data as well as watermark
data and optimises the embedding LUT. We use image as an
example without loss of generality. Since our scheme is not
targeting for particular watermarking applications, data
hiding and watermarking are used interchangeably in the
context. Notations in this paper are shown in Table 1.

### Table 1 Table of nomenclature

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q(·)</td>
<td>quantisation function</td>
</tr>
<tr>
<td>T</td>
<td>LUT</td>
</tr>
<tr>
<td>tk</td>
<td>the k-th entry of a LUT T</td>
</tr>
<tr>
<td>T</td>
<td>0 if tk = 1, 1 if tk = 0</td>
</tr>
<tr>
<td>b</td>
<td>bit to be embedded</td>
</tr>
<tr>
<td>sb</td>
<td>host feature/coefficient to be embedded b</td>
</tr>
<tr>
<td>f(sb)</td>
<td>probability density of sb</td>
</tr>
<tr>
<td>DistT</td>
<td>mean square quantisation introduced distortion of the k-th cell</td>
</tr>
<tr>
<td>DistTk</td>
<td>mean square LUT embedding introduced distortion of the k-th cell</td>
</tr>
<tr>
<td>MSEquan</td>
<td>overall quantisation mean square distortion</td>
</tr>
<tr>
<td>MSEw(T)</td>
<td>overall LUT embedding distortion when T is used</td>
</tr>
<tr>
<td>Dl(T)</td>
<td>extra LUT embedding distortion when T is used</td>
</tr>
<tr>
<td>Pl(T)</td>
<td>probability that a feature is mapped to the k-th entry after LUT embedding</td>
</tr>
<tr>
<td>Rl(T)</td>
<td>robustness contribution of the k-th entry when T is used</td>
</tr>
</tbody>
</table>

### 2 Overview of the LUT embedding

An LUT T is a sequence of 0s and 1s, associated with a
uniform quantiser. It maps every feature or pixel value of
an image to a quantisation level according to the input data
to be embedded. The embedding and detection process of
LUT-based data hiding is shown in Fig. 1.

First the host elements are quantised. A uniform quantiser
with cell width q maps the original host signal to kq,
k = 1, . . . , K, where K is the size of the LUT. Note that here
we assume that the signal value has already been normalised
to be positive without loss of generality. Each quantiser cell,
kq, carries an information bit that is represented by the
corresponding k-th entry in the LUT. If one bit is to be
embedded into a host coefficient, the coefficient is mapped to
the nearest quantisation value whose corresponding LUT
entry is the same as the information bit. For example to
embed a ‘1’ in a pixel, the pixel is rounded by its
quantisation value if the entry of the table corresponding to
that pixel is also a ‘1’. If the entry is not ‘1’, we should find
its nearest quantisation level for which its LUT entry is ‘1’
to replace the pixel as illustrated in Fig. 2. The process of
embedding ‘0’ is the same.

The look-up function Lookup(·) simply returns a ‘0’ or ‘1’
depending upon the input index.

\[
\text{Lookup}(f) = \text{value in LUT at index } I
\]

![Fig. 1 Diagram of the LUT-based data hiding](image-url)
The LUT(·) function takes the value of the original signal as the input and maps it to a ‘0’ or ‘1’ according to the LUT. Thus, the LUT(·) function is actually a simple composition of the look-up and the quantisation functions

\[
\text{LUT}(s) = \text{Lookup}(Q(s)/q)
\]

where \( q \) is quantisation step and \( Q \) is quantisation function.

The entire process altering a pixel can be abstracted into the following formula

\[
x = \begin{cases} 
Q(s) & \text{if } \text{LUT}(s) = b \\
s + \delta & \text{if } \text{LUT}(s) \neq b 
\end{cases}
\]

where \( s \) is the original feature (in this case, pixel value), \( x \) is the watermarked feature, \( b \) is the bit to be embedded and \( \delta = \arg \min_{\{|d|\}} |d| = (Q(s) - s) \) s.t. LUT \((z) = b\).

Once the LUT is known, the watermark detection can be easily implemented through a simple look-up from the LUT. The table is looked up as

\[
\hat{b} = \text{LUT}(\hat{x})
\]

where \( \hat{b} \) is the extracted bit and \( \hat{x} \) is the watermarked, possibly corrupted signal.

A typical LUT embedding algorithm is the odd–even embedding. First, a uniform quantiser \( Q(\cdot) \) is defined. The partition of the quantiser is shown in Fig. 3. The host pixel is mapped to the nearest even number point to embed a ‘0’ and the nearest odd number quantisation point to embed a ‘1’. Thus, a relationship between the information bit and the marked signal is formed. In this scheme, the LUT entries for embedding ‘0’ and ‘1’ are arranged in an interleaving manner. It is formulated as LUT of run length 1 in [16]. It is also noted that the LUT with larger run constraints introduces larger distortion but has better robustness and thus smaller BER. In this paper, our goal is to design LUT so as to minimise distortion while keeping the run constraints unchanged.

3 Distortion analysis

In LUT embedding, uniform quantisation \( Q(\cdot) \) divides the input signal space into \( K \) levels. If the \( l \)th entry of LUT is \( b \), to embed \( b \) the data samples of signal \( s \) in the quantisation cell of \([k - 1/2]q, (k + 1/2]q\) is rounded to \( kq \), the mean square distortion produced by this operation is calculated as

\[
D_k(s_b) = \int_{(k-1/2)q}^{(k+1/2)q} |s - kq|^2 f(s) \, ds
\]

where \( f(s) \) is the probability density function (PDF) of \( s \) and the features to embed bit ‘\( b \)’ is denoted by \( s_b \). However, if the bit to be embedded for \( s \) is not \( b \), the host data must be mapped to the nearest quantisation point corresponding to the desired bit. There are three cases:

- The \((k + l)\)th entry is the only closest entry for the desired bit;
- The \((k - l)\)th entry is the only closest entry for the desired bit;
- Both the \((k + 2)\)th and \((k - 2)\)th entries are the closest entries.

An example of \( l = 2 \) is illustrated in Fig. 4.

If the \((k + l)\)th entry is the only closest entry for the desired bit [Fig. 4a], the distortion of the \( k \)th entry is

\[
D_k(s_b) + \delta^2 q^2 \int_{(k-1/2)q}^{(k+1/2)q} f(s) \, ds - 2\delta q \int_{(k-1/2)q}^{(k+1/2)q} s f(s) \, ds
\]

If the feature is approximately symmetric distributed within each cell, the last term is close to 0. We have

\[
D_k(s_b) \approx D_k(s_b) + \delta^2 q^2 \int_{(k-1/2)q}^{(k+1/2)q} s f(s) \, ds
\]

Similarly, if the \((k + l)\)th entry is the only one closest entry

\[
D_k(s_b) \approx D_k(s_b) + \delta^2 q^2 \int_{(k-1/2)q}^{(k+1/2)q} s f(s) \, ds
\]

3 Distortion analysis

In LUT embedding, uniform quantisation \( Q(\cdot) \) divides the input signal space into \( K \) levels. If the \( l \)th entry of LUT is \( b \), to embed \( b \) the data samples of signal \( s \) in the quantisation cell of \([k - 1/2]q, (k + 1/2]q\) is rounded to \( kq \), the mean square distortion produced by this operation is calculated as

\[
D_k(s_b) = \int_{(k-1/2)q}^{(k+1/2)q} |s - kq|^2 f(s) \, ds
\]

where \( f(s) \) is the probability density function (PDF) of \( s \) and the features to embed bit ‘\( b \)’ is denoted by \( s_b \). However, if the bit to be embedded for \( s \) is not \( b \), the host data must be mapped to the nearest quantisation point corresponding to the desired bit. There are three cases:

- The \((k + l)\)th entry is the only closest entry for the desired bit;
- The \((k - l)\)th entry is the only closest entry for the desired bit;
- Both the \((k + 2)\)th and \((k - 2)\)th entries are the closest entries.

An example of \( l = 2 \) is illustrated in Fig. 4.

If the \((k + l)\)th entry is the only closest entry for the desired bit [Fig. 4a], the distortion of the \( k \)th entry is

\[
D_k(s_b) + \delta^2 q^2 \int_{(k-1/2)q}^{(k+1/2)q} f(s) \, ds - 2\delta q \int_{(k-1/2)q}^{(k+1/2)q} s f(s) \, ds
\]

If the feature is approximately symmetric distributed within each cell, the last term is close to 0. We have

\[
D_k(s_b) \approx D_k(s_b) + \delta^2 q^2 \int_{(k-1/2)q}^{(k+1/2)q} s f(s) \, ds
\]

Similarly, if the \((k + l)\)th entry is the only one closest entry

\[
D_k(s_b) \approx D_k(s_b) + \delta^2 q^2 \int_{(k-1/2)q}^{(k+1/2)q} s f(s) \, ds
\]
for the desired bit [Fig. 4b], the distortion is

\[
D_{k}^{\pm} = \int_{(k-1)/2q}^{kq} |s - (k - l)q|^2 f(s) ds
\]

where \( T \) is LUT, \( \alpha \) and \( \beta \) are calculated as follows

\[
\alpha_{0,k} = \max \{ t_{k-1}, t_{k+1}, \ldots, t_{k-q-1}, t_{k+q-1}, t_{k-q}, t_{k+q}, \ldots, t_{k} \}
\]

\[
\alpha_{1,k} = \max \{ t_{k-1}, t_{k+1}, \ldots, t_{k-q-1}, t_{k+q-1}, t_{k-q}, t_{k+q}, \ldots, t_{k} \}
\]

\[
\beta_{0,k} = \alpha_{0,k} - t_{k+1}
\]

\[
\beta_{1,k} = \alpha_{1,k} - t_{k+1}
\]

\[\beta_{b,k} = 1 \text{ only when } t_{k} = b, t_{k-1} = b \text{ or } t_{k+1} = b \text{ and is the nearest LUT entry for } b. \]

\[\beta_{b,k} = 0 \text{ otherwise.}
\]

\[\text{From (11), the overall distortion can be formulated as}
\]

\[
\text{MSE}_{w}(T) = \sum_{k=0}^{K-1} \text{Dist}_{k}(T)
\]

\[\text{The overall embedding distortion is}
\]

\[
\text{MSE}_{w}(T) = \text{MSE}_{\text{quan}} + \sum_{k=0}^{K-1} D_{k}(T)
\]

\[\text{4 Minimum distortion LUT with VA}
\]

The overall structure of the proposed data hiding scheme is illustrated in Fig. 5 for binary case. First, host elements are quantised using a uniform quantiser. A VA is used in order to find the optimal distortion LUT according to the watermark signal and the quantised host data described in the following paragraph. Then the optimal LUT is used as a key to quantise the host data. After that the watermarked data is transmitted through the channel. The channel could be any kind of attack or noise. At the detector side the LUT is used as a secret key in order to find the watermark inside the received data. The proposed method is a blind detection
method which does not need the original image at the receiver side.

Since \( \text{MSE}_{\text{quan}} \) is the same for all \( n \)-run LUTs, that is, for all \( T \) in (17), the optimal \( n \)-run LUT is the one that minimises the additional distortion

\[
T_{\text{opt}} = \arg \min_T \left\{ \sum_{k=0}^{K-1} D_k(T) \right\} \tag{18}
\]

We formulate it as a problem of minimising \( K \) steps summation of \( D_k(T) \) and it can be solved using dynamic programming. A VA [18] is used. For a \( n \)-run LUT, \( 2^{2n-2} \) states is represented by the \( 2n-2 \) neighbouring LUT entries

\[
S = \{ t_{k-n+1} \cdots t_k \cdots t_{k+n-2} \}. \tag{19}
\]

In each state of the trellis, the previous state metric (SM) and the corresponding branch metric (BM) are added together, and then the accumulated SM is updated by choosing the minimum of all possible cases recursively

\[
\text{SM}^k_{S_i} = \min(\text{SM}^k_{S_j} + \text{BM}^{k+1}_{S_j}, k = 0, \ldots, K - 2) \tag{20}
\]

where \( \text{SM}^k_{S_i} \) represents the SM of the \( j \)th state at step \( k \), and \( \text{BM}^k_{S_i} \) denotes the BM at step \( k \) associated with a transition from state \( S_j \) to state \( S_i \). A transition happens only when the last \( 2n-3 \) entries of \( S_j \) is the same as the first \( 2n-3 \) entries of \( S_i \). Fig. 6 shows the trellis of a 2-run LUT.

The initial state metric \( \text{SM}^0_{S_i} \) is given by the additional distortion of \( k = 0 \).

\[
\text{SM}^0_{S_i} = D_0(S_j) \tag{21}
\]

where \( D_0(S_i) \) denotes the additional distortion, whereas \( t_{k-n+1}, \ldots, t_k, \ldots, t_{k+n-1} \) are given by \( S_j \).

Let \( \text{BM}^k_{S_i,S_j} \) be the additional distortion of the \( k \)th entry. Since the additional distortion of the \( k \)th entry is decided only by the \( 2n-1 \) nearby entries \( t_{k-n+1}, \ldots, t_k, \ldots, t_{k+n-1} \) which can be obtained from \( S_j \) and \( S_i \). Considering the case that from \( S_j \) to \( S_i \) will break the run \( n \) constrain, \( \text{BM}^k_{S_i,S_j} \) is given by a modification of the additional distortion.

\[
\text{BM}^k_{S_i,S_j} = \begin{cases} 
  D_k(S_j, S_i) & \text{if run } > n \text{ from } S_j \text{ to } S_i \\
  \text{else} & 
\end{cases} \tag{22}
\]

where \( D_k(S_j, S_i) \) denotes the additional distortion while \( t_{k-n+1}, \ldots, t_k, \ldots, t_{k+n-1} \) are given by \( S_j \) and \( S_i \). Then the accumulated SM is the overall additional distortion for all the \( K \) entries. We can create the minimum distortion \( n \)-run LUT using the corresponding state path that minimises the accumulated SM.

The algorithm could be summarised in the following steps:

- **Step 1:** Calculate the additional distortion of selecting each path BM.
- **Step 2:** Add the previous SM and the corresponding BM together and select the less distortion path of each state.
- **Step 3:** Find the minimum distortion LUT by choosing the minimum accumulated SM which is the overall additional distortion for all \( K \) entries.

The complexity of the new LUT design algorithm is similar to that of the regular VA. The complexity increases linearly with the number of quantisation levels. For example, if 3-run LUT with 20 quantisation levels is used, the complexity of both embedding and decoding is the same as eight states length 20 trellis decoding. Compared against our previous reduced distortion method [17], there is no complexity increase as the reduced distortion method needs more comparison and ranking. Compare to odd–even method which uses a simple odd–even encoding, the complexity increase is the VA. But the performance is greatly increased over the odd–even method. Also, note that all the computational complexity is in the LUT design process at the embedding end. Once the LUT has been designed, there is minimal computation cost in both watermark embedding and decoding.

### 5 Practical data hiding in the wavelet domain

In the real image data hiding case, it is preferable to embed the information in the transform domain [19]. A new scheme that selects the large coefficients based on a Gaussian mixture model in the wavelet domain is explored in our experiment. In general, wavelet coefficients with large magnitude could survive the basic image processing and compression attacks. The wavelet coefficients could be modelled as a two-component Gaussian mixture, since the
Wavelet coefficients have a peaky, heavy-tailed marginal distribution \([20, 21]\) and a near-zero mean. One component includes large coefficients which are singularities such as edges. The other takes small values. This statistical feature can be expressed by using a two-component Gaussian mixture model

\[
p(o_i) = p_s g(o_i, 0, \delta_s^2) + p_l g(o_i, 0, \delta_l^2)
\]

\[
p_s + p_l = 1
\]

where the small coefficient component is represented by subscript ‘s’ and the large by ‘l’. The priori probabilities of the two are \(p_s\) and \(p_l\), respectively. The variances are \(\delta_s\) and \(\delta_l\). The parameters can be calculated by using the expectation-maximisation (EM) algorithm.

We use the Gaussian mixture model (GMM) to find large coefficients for hiding data. The information bits are only embedded into the large wavelet coefficients. The number is \(m p_l\).

6 Experimental results with images

To verify our scheme, the proposed minimum distortion LUT embedding with run constraints of 2 and 3 is applied to 20 popular \(512 \times 512\) images of different types including Lena, Bridge and Goldhill in the spatial domain. The 20 images are shown in Fig. 7.

For comparison purposes, given the embedding rate as one bit per pixel, the performance of the embedding scheme using the odd–even LUT (i.e. the LUT with run of 1) and the average performance of LUTs with a given run are calculated. Note that the average performance of LUTs with a given run is calculated by 100 randomly generated LUTs under same run length constraints.

We also calculated the average performances of all possible 2-run and 3-run LUTs, respectively. Fig. 8 shows the PSNR (peak-signal-to-noise-ratio) comparison for odd–even LUT, the minimum distortion LUT, the reduced distortion LUT [17] and the average performance of the LUT embedding at different quantisation levels. As can be seen, PSNR of the new minimum distortion LUT embedding is the best at all

![Fig. 7 Twenty popular 512 × 512 images of different types](image)
levels, although in general longer runs are expected to generate worse PSNR than odd–even LUT embedding. The new minimum distortion embedding is also much better than the average performance of the LUT embedding. When the number of quantisation level increases, the difference gets smaller. The underlying reason is that the distortion of the odd–even LUT embedding gets smaller with more quantisation levels and leaves less space for improvement. It is also shown that our new minimum distortion LUT method has significant improvement over our previous reduced distortion LUT method [17], although the reduced distortion LUT does a much better job than the odd–even embedding and average 2-run LUT.

Next, we add white Gaussian noise to watermarked images with the minimum distortion LUT, the odd–even LUT and the average LUT. The detection errors on $512 \times 512$-bit raw data at different WNR are shown in Fig. 9. Fig. 10 visualises the detection errors from which we can note the minimum distortion LUT has a great improvement on reducing the raw BER. The PSNR is also increased from 26.54 dB of odd–even to 27.30 dB of minimum distortion LUT.

The 3-run LUT only has slight improvement over the 2-run LUT in Fig. 9 because the distributions of the watermark data and the host data are such that the optimal 3-run LUT does not contain many 3 runs and the distortions of 2 runs dominate. As a special case, a watermark signal with 98% 0s and 2% 1s is also tested in our experiment, which often happens if a binary text watermark image is embedded. The WNR against BER performance is shown in Fig. 11. The 3-run LUT is about 1 dB better than 2-run LUT. It means the long run
LUT will do better with uneven embedding. Also, it is shown that the robustness performance of the reduced distortion LUT is almost the same as the minimum distortion 2-run LUT as expected as the run length is correlated to the robustness.

Finally, we test our new practical data hiding in the wavelet domain. Fig. 12 shows the PSNR after embedding using our new scheme. It is the same as in the spatial domain PSNR of the minimum distortion LUT embedding is better than other LUT embedding methods at all levels. Fig. 13 shows the performance of new method against the JPEG attack. Our results are not designed specially for watermarking. Attacks are not tested thoroughly. Anyway, it can be seen from the results that our new scheme is suitable for data hiding applications that need to achieve less distortion at certain level of robustness. Also note that new minimum distortion LUT has much better PSNR (distortion) and a slightly better robustness compared to the reduced distortion LUT.

Note that the main target of the algorithm is for high payload data hiding. Our embedding data rate is 1 bit/pixel or 1 bit/coefficient in the wavelet case which is much greater than 1 bit/image in spread spectrum watermarking for copyright protection. The malicious attacks of watermarking are not concerns in such data hiding applications and are therefore not tested.

The presented data hiding method is a host data- and watermark data-dependent method. It takes advantage of the distribution information of host data as well as watermark data and optimises the embedding LUT. The method does not depend on specific properties of images. It can therefore be applied to other types of data such as audio and video, etc. As long as the host data are not uniformly distributed, the optimised embedding LUT is better than the conventional odd–even LUT.

7 Conclusion

In this paper, a new optimal LUT data hiding scheme is presented to minimise the mean square distortion given certain robustness represented by the length of the run. Through the distortion analysis, we generalise the embedding distortion function and formulate the distortion minimisation problem as a dynamic programming problem. A VA is employed to find the minimum distortion LUT. Some practical considerations are also discussed. Experimental results show that our presented scheme with a run constraint larger than 1 is more robust and has less distortion than traditional LUT embedding schemes such as odd–even LUT embedding in both transform and spatial domains. The presented embedding scheme is distinguished by its ability to achieve minimum distortion according to the distribution of the watermark signal. In practice, more than one near minimum distortion LUTs can be generated by choosing alternative paths to enhance the embedding security, because it makes it difficult to drive the LUT used for data hiding even if the original host data are known. The presented algorithm is suitable to optimise joint robustness, fidelity and security. Future work may include exploring optimal LUT performances that suits the requirements of human visual systems.

8 References